

École polytechnique de Louvain

Automated Tracking and Re-Identification of Chimpanzees in Videos

Authors: **Thomas RIXEN, Diego TROCH**

Supervisor: **Yves DEVILLE**

Readers: **Yves DEVILLE, Théodore COUSIN, Calogero MONTEDORO,
Christophe DE VLEESCHOUWER**

Academic year 2025–2026

Master [120] in Computer Science

Abstract

This thesis presents the development of a computer vision architecture to automatically detect, track, and recognise individual chimpanzees in video footage. The project was designed to support the doctoral research of Calogero Montedoro, which studies how environmental conditions influence the social development of juvenile chimpanzees. By reducing the amount of manual identification required to process long videos, the goal is to make behavioural studies more scalable and less time-consuming.

The proposed system is modular and combines body detection, face detection, multi-object tracking, and identity assignment. It builds on YOLOv8 for body detection, ByteTrack for motion-based tracking, and a ChimpUFE-based face embedder fine-tuned for the study group. In addition, this thesis introduces a three-step annotation workflow and a HOTA-based evaluation pipeline to support both dataset production and reproducible comparison between methods.

To improve recognition, a three-tier face dataset was assembled from curated identity images, student annotations, and automatically mined face crops from annotated videos. Building on this model, an auto-labelling pipeline was developed to assign names to tracklets and refine them through clustering and temporal post-processing.

Although some limitations remain, especially under long occlusions, dense crossings, and difficult viewing conditions, the resulting pipeline can already be used in a semi-automatic way. Corrected videos and new annotations can in turn be reused to strengthen the training data and further improve the system in future work.

Acknowledgments

We would first like to thank Professor Yves Deville for his guidance, regular follow-up, and valuable advice throughout the academic year.

We are also deeply grateful to Théodore Cousin, our co-supervisor, for his technical support, implementation suggestions, and experience from the previous year of the project.

We warmly thank Calogero Montedoro for helping define the biological objectives of the project and for his contribution to the annotation effort. We are also grateful to Brigitte Rezsöházy for her extensive video annotation work, which was central to the construction of the ground-truth corpus.

We would like to thank Professor Christophe De Vleeschouwer for his valuable advice on multi-object tracking and for the discussions that helped refine several technical choices of this work.

Finally, we thank all the people who contributed, directly or indirectly, to the project, the annotation workflow, and the experimental infrastructure.

Computational resources were provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

Preface

Use of Artificial Intelligence

Artificial intelligence tools, in particular large language models, were used as supportive instruments for writing, literature exploration, programming, and figure production. They helped us clarify wording, improve consistency, explore technical options, debug code, and script some programmatic figures, but they did not replace the scientific reasoning, methodological choices, implementation decisions, or interpretation of the results.

During writing and literature exploration, these tools were used for translation, linguistic refinement, sentence reformulation, and preliminary explanations or summaries of technical papers. Any scientific claim or bibliographic reference included in the thesis was checked against the original source.

During implementation, AI-assisted code or figure suggestions were reviewed, adapted, and tested before being integrated into the project. The scientific content of the thesis, including the problem formulation, experimental design, implementation choices, analysis, and conclusions, remains our own responsibility.

The main tools used were ChatGPT, GitHub Copilot, NotebookLM, DeepL, and Manim.

Contents

Acknowledgments	i
Preface	ii
1 Introduction	1
1.1 Context and Motivation	1
1.2 Problem Statement	2
1.3 Approach and Contributions	2
1.4 Structure of the Work	3
2 Background	4
2.1 Detection Methods	4
2.1.1 YOLO: Single-Stage Object Detection	4
2.2 Multi-Object Tracking (MOT)	5
2.2.1 General Concepts	5
2.2.2 DeepSORT	6
2.2.3 StrongSORT	7
2.2.4 ByteTrack	9
2.2.5 BoT-SORT	11
2.2.6 Comparison on MOT17	14
2.3 Re-ID models	15
2.3.1 OSNet: Omni-Scale Feature Learning	15
2.3.2 ChimpUFE	16
2.3.3 ArcFace: Angular Margin Loss for Face Recognition	18
2.4 Evaluation Metrics for MOT	19
2.4.1 Re-Identification Metrics: CMC and mAP	19
2.4.2 MOTA	20
2.4.3 IDF1	21
2.4.4 HOTA	21
2.4.5 Choice of Primary Metric	26
2.5 The ChimpRec Pipeline: Starting Point	26
2.5.1 Architecture	26
2.5.2 Components and Reported Performance	27
2.5.3 Limitations Motivating This Thesis	28

3	Datasets and Evaluation Infrastructure	29
3.1	Manual Annotation Workflow	29
3.1.1	Three-Step Rendering Pipeline	30
3.2	Ground-Truth Video Corpus	31
3.2.1	Filming Setup and Camera Motion	31
3.2.2	Selection of Evaluation Videos	32
3.3	Preparing Data for HOTA Evaluation	33
3.3.1	A Single Interchange Format for Ground Truth and Predictions	33
3.3.2	From Text Files to HOTA Inputs	34
3.3.3	Identity-Aware Recognition Metrics	36
3.4	Body Re-Identification Dataset	37
3.5	Face Embedder Fine-Tuning	37
3.5.1	Motivation	37
3.5.2	Three-Tier Dataset Overview	38
3.5.3	Tier 1 ChimpRec Database	38
3.5.4	Tier 2 Student Trombinoscope	39
3.5.5	Tier 3 - Mined Face Crops from Annotated Videos	39
3.5.6	Test Set	41
4	Methods	43
4.1	Experimental Hardware and Runtime Environment	43
4.2	Tracking Methods	43
4.2.1	Common Tracking Protocol	43
4.2.2	Inherited DeepSORT Baseline	44
4.2.3	StrongSORT Integration	44
4.2.4	ByteTrack as a Motion-Only Tracker	44
4.2.5	BoT-SORT Re-ID Ablation	45
4.3	Fine-Tuning ChimpUFE with an ArcFace Identity Head	45
4.3.1	Overview	45
4.3.2	Backbone and Identity Head	46
4.3.3	Two-Stage Fine-Tuning Schedule	46
4.3.4	Tier-Aware Sampling and Loss Weighting	47
4.3.5	Per-Tier Data Augmentation	48
4.3.6	Evaluating the Impact of Tier 3 Data	49
4.3.7	Validation Protocol	49
4.4	Post-Processing Methods	50
4.4.1	Dual-Heuristic Spatio-Temporal Post-Processing	50
4.4.2	Face-Recognition-Driven Auto-Labelling Pipeline	52
5	Results and Analysis	59
5.1	Experimental Protocol and Metric Caveats	59
5.2	Face Embedder Fine-Tuning Results	60
5.2.1	Training Dynamics	60

5.2.2	Test-Set Identification Performance	60
5.2.3	Per-Identity Behavior and Failure Modes	62
5.2.4	Effect of Tier 3 Data and Training-Set Size	64
5.2.5	Embedding Geometry	64
5.2.6	Confidence-Based Abstention and Operating Points	66
5.3	Tracker Methods Comparison	67
5.3.1	Evaluation Protocol and Ground-Truth Caveat	67
5.3.2	Hyperparameter Tuning on the Tuning Videos	67
5.3.3	Comparison on the Evaluation Videos	68
5.4	Post-Processing Results	69
5.4.1	Post-Processing on the Evaluation Videos	70
5.4.2	Recognition Accuracy of the Final Pipeline	71
5.4.3	Residual Errors	73
5.5	Discussion	73
6	Future Work	76
6.1	Data and Annotation	76
6.2	Modelling	77
6.3	Pipeline-Level Refinements	78
6.4	Behavioural Downstream Tasks	79
7	Conclusion	80
7.1	Summary of the Thesis	80
7.2	Main Contributions	81
7.3	Main Findings	82
7.4	Limitations and Future Directions	82
7.5	Final Perspective	83
A	Keypoint-Based Approach	84
A.1	Motivation	84
A.2	Data and Training	84
A.2.1	Skeleton Design and Annotation	84
A.2.2	Training and Inference	85
A.3	Why the Pipeline Was Abandoned	86
B	Tracker Hyperparameter Search Details	87
B.1	StrongSORT	87
B.2	BoT-SORT	88
B.3	ByteTrack	89
B.4	Selected Configurations	90
C	Auto-Labelling Pipeline Hyperparameters	92
C.1	Post Processing Selected Configurations	92

D Tier Statistics and Tier-Aware Contribution Computation	94
D.1 Expected Tier Contribution Computation	95
D.1.1 Sampling probabilities after resampling cap	95
D.1.2 Global resampling cap	96
D.1.3 Expected loss-weighted contributions	96
E Additional Face-Model Diagnostics	98
E.1 Confusion Matrix Face Recognition Models	98
E.2 Training-Set Size Diagnostic	98
E.3 Model Agreement	99

Chapter 1

Introduction

1.1 Context and Motivation

Behavioural analysis from chimpanzee video depends on knowing which individual appears where, and when. Before recordings can support analyses of proximity, co-occurrence, social interaction, or play behaviour, the animals visible in each scene must be tracked through time and associated with their identities. In practice, this annotation step remains a major bottleneck, long recordings require substantial inspection, correction, and interpretation.

This work was developed in support of the doctoral research of Calogero Montedoro, which studies how environmental conditions influence the social development of juvenile chimpanzees. The biological objective is not only to know that several chimpanzees are present in a frame, but to recover identity-aware trajectories that can later be transformed into behavioural variables. Manual annotation remains possible on short fragments, but it scales poorly when the footage contains many individuals, frequent interactions, and long periods of ambiguous visibility.

The present thesis continues the **ChimpRec** project introduced by Cousin and Demeure [1]. Their work provided the first modular pipeline for the study group, including YOLOv8 body and face detectors, a DeepSORT-based tracking stage, and a manual correction interface. It also revealed the main weakness of the system, the detector could often localise chimpanzees, but the pipeline could not reliably maintain or assign identities without substantial human correction. The question addressed in this thesis is therefore more specific than building a detector. It is how to turn fragmented, anonymous body tracklets into temporally coherent trajectories carrying useful chimpanzee names.

This setting is challenging for several reasons. The videos show groups of young chimpanzees in captive environments, often during play or close social contact. Individuals have similar global body appearance, change posture rapidly, and frequently occlude one another. The most discriminative identity cues are facial, but faces are visible only intermittently and may be blurred, turned away, partially hidden, or too small in the frame. The camera is also handheld, with pans, jitter, and optical zoom. A practical annotation system must therefore combine robust body localisation with identity evidence that is sparse, noisy, and unevenly distributed across time.

1.2 Problem Statement

The core problem addressed in this thesis is semi-automatic, identity-aware annotation of chimpanzee video. Given an input sequence, the system should produce per-frame body boxes that remain consistent over time and, whenever enough evidence is available, carry the correct identity among the known individuals of the study group.

This formulation separates two notions that are easy to conflate. A multi-object tracker only needs to maintain arbitrary trajectory identifiers, such as **Track 17**, consistently through time. A behavioural annotation, however, needs the literal chimpanzee identity, such as **Kassongo** or **Kira**. A method can score well on trajectory association while still assigning the wrong biological name, and conversely a name can be locally correct while the trajectory remains fragmented. For that reason, this thesis evaluates both tracking quality, mainly through HOTA [2], and identity-aware naming quality.

The working hypothesis of the thesis is that this task should not be solved as a purely online tracking problem. Body detections are available in most frames and are suitable for short-term motion tracking, but body appearance is too weak to distinguish juvenile chimpanzees reliably. Face evidence is much more discriminative, but it appears only in a subset of frames. The system is therefore designed in two stages, first produce body tracklets with a tracking-by-detection method, then assign identities offline by accumulating face evidence over those tracklets.

1.3 Approach and Contributions

The thesis investigates this two-stage design through a sequence of datasets, models, and evaluation tools. Its main contributions are the following.

- A three-step annotation workflow was built to generate raw tracker outputs, manually corrected identity annotations, and clean final renderings for collaborators.
- An eleven-video ground-truth corpus was assembled, covering the study group and providing separate training, tuning, and evaluation material.
- A HOTA-compatible evaluation layer was implemented for the project-specific #-delimited tracking format, together with complementary identity-aware metrics for the final named output.
- Several online trackers were compared under a shared cached-replay protocol, including the inherited DeepSORT baseline, StrongSORT, ByteTrack, and BoT-SORT with and without chimpanzee body Re-ID.
- A three-tier face dataset was constructed from curated identity images, student annotations, and automatically mined video crops, then used to fine-tune ChimpUFE [3] with an ArcFace identity head.
- Two offline post-processing strategies were developed. First an unsupervised dual-heuristic clustering baseline and then a final face-recognition-driven auto-labelling pipeline that names, merges, repairs, and interpolates tracklets.

These contributions also include several negative or diagnostic results that shaped the final design.

1.4 Structure of the Work

The remainder of this thesis is organised as follows. Chapter 2 reviews the detection, tracking, Re-ID, face-recognition, and evaluation concepts used in the work, and positions the thesis relative to the previous **ChimpRec** pipeline. Chapter 3 describes the annotation workflow, ground-truth corpus, HOTA evaluation infrastructure, body Re-ID dataset, and three-tier face dataset. Chapter 4 details the tracking protocols, ChimpUFE fine-tuning procedure, and post-processing pipelines. Chapter 5 reports the quantitative and qualitative results. Chapter 6 discusses concrete extensions of the system. Finally, Chapter 7 summarises the main findings and practical impact of the thesis.

The implementation underlying this work is open source and available on GitHub at <https://github.com/diegRock/ChimpRec>.

Chapter 2

Background

Chapter Overview. This chapter first reviews the detection and tracking-by-detection methods considered in our experiments, followed by the Re-ID models used as appearance encoders. It then introduces the evaluation metrics used throughout the thesis, covering both Re-ID retrieval metrics and multi-object tracking metrics, with emphasis on HOTA as our primary MOT criterion. The chapter closes with a summary of the legacy ChimpRec pipeline, whose limitations motivate the contributions of this thesis.

2.1 Detection Methods

2.1.1 YOLO: Single-Stage Object Detection

YOLO (You Only Look Once) [4] reformulated object detection as a single regression problem mapping image pixels directly to bounding boxes and class probabilities in one forward pass.

Main idea

YOLO divides the input image into a fixed $S \times S$ grid. Each grid cell is responsible for detecting objects whose center falls inside it. For every cell the network predicts a small number of bounding boxes (each with its center coordinates, width and height), a confidence score reflecting both object presence and localization quality, and a set of class probabilities shared across the boxes of the cell. Class-specific confidence scores are then obtained at inference time by multiplying the box confidence with the class probabilities. A non-maximum suppression step removes overlapping predictions by keeping only the highest-confidence box for each object.

Network Architecture and Training

During training, YOLO uses a loss function that combines localisation, objectness, and classification errors. The localization loss penalizes deviations in predicted bounding box coordinates and dimensions, the objectness loss encourages the model to assign high confidence to cells containing objects and low confidence to empty cells. The classification loss ensures that the predicted class probabilities align with the ground truth labels for detected objects.[4]

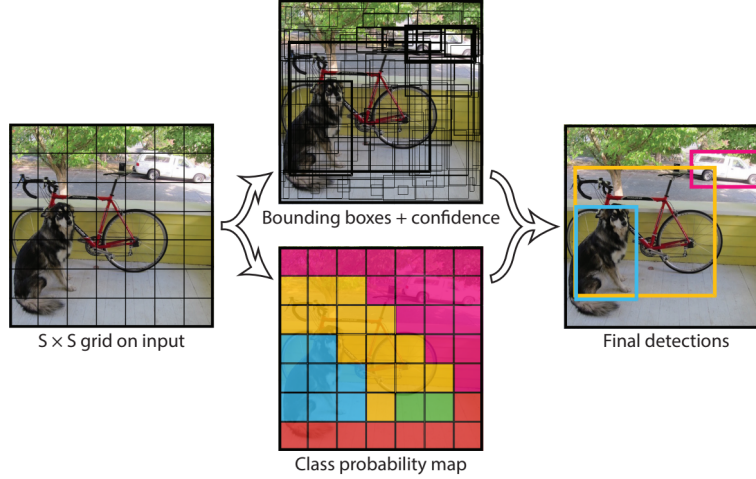


Figure 2.1: Illustration of YOLO modeling detection as a regression problem. [4]

YOLOX & YOLOv8

Newer YOLO variants such as YOLOX [5] and YOLOv8 [6] were introduced to address limitations of earlier YOLO versions, especially the strong constraints imposed by grid-based prediction and the need to tune anchor boxes for each dataset. They move toward simpler and more flexible detection formulations described as *anchor-free* and improve the detection head design by separating classification from box regression, which tends to help localisation and classification simultaneously.

2.2 Multi-Object Tracking (MOT)

2.2.1 General Concepts

All four trackers compared in this thesis follow the *tracking-by-detection* paradigm. A frame-level detector (e.g. *YOLO*) outputs independent bounding boxes at every frame, and the tracker is then in charge of linking those frame-level boxes into trajectories.

Three main concepts recur in every tracker described below, so we introduce them once here.

Kalman filter (motion prediction). A Kalman filter [7] is a recursive estimator that maintains, for each tracked object, a *state* (its position, size and velocities in image space) and an associated uncertainty. At each new frame it produces a prediction of where the object should be, and it updates this prediction once a new detection is associated with the track.

Hungarian algorithm (data association). Once a tracker has predicted where each existing track should be in the new frame, it must decide which detection corresponds to which track. This is a bipartite assignment problem. For every (track, detection) pair, the tracker computes a cost, and the Hungarian algorithm [8] finds the assignment that minimises the total cost subject to the constraint that each track is matched to at most one detection and vice versa. The trackers below differ mostly in how they build this cost matrix.

Intersection over Union (IoU). The simplest cost between a predicted box and a candidate

detection is $1 - \text{IoU}$, where $\text{IoU} = |A \cap B| / |A \cup B|$ measures the overlap area divided by the union area of the two boxes. IoU is dimensionless and lies in $[0, 1]$, with 1 meaning perfect overlap.

The four subsections below describe how DeepSORT, StrongSORT, ByteTrack and BoT-SORT instantiate this template, focusing on what is specific to each one.

2.2.2 DeepSORT

DeepSORT [9] (*Simple Online and Realtime Tracking with a Deep Association Metric*) extends SORT [10] by adding an appearance term to the association cost. SORT relied only on motion overlap, and tended to switch identities whenever two objects came close or one of them disappeared for a few frames. The appearance term is meant to bridge those gaps.

Motion Model

Each track stores an eight-dimensional Kalman state $(u, v, \gamma, h, \dot{u}, \dot{v}, \dot{\gamma}, \dot{h})$, where (u, v) is the bounding-box center in image coordinates, γ the aspect ratio, h the height, and the dotted terms the corresponding velocities. The formulation lives in image space and assumes nothing about the camera. A track is kept alive as long as it receives regular updates and is otherwise deleted after a maximum age. Detections that do not match any track open a new *tentative* track, which is only confirmed once it has been associated to detections in several consecutive frames [9].

Detection-to-Track Association

For every (track, detection) pair, DeepSORT builds a cost from two complementary terms. The motion term is the *Mahalanobis distance* between the predicted Kalman state of the track and the candidate detection. Compared to a plain Euclidean distance, the Mahalanobis distance scales each axis by the corresponding Kalman uncertainty. A detection that lies one pixel away from a confident prediction counts more than one that lies one pixel away from an uncertain prediction. Pairs whose Mahalanobis distance falls outside a χ^2 confidence region are discarded (*gating*).

The appearance term is a cosine distance between feature vectors. A small CNN maps each detection to a normalised 128-dimensional embedding, and each track keeps a small *gallery* of its most recent embeddings. The track-to-detection appearance distance is the minimum cosine distance between the candidate embedding and any embedding in the gallery, which makes the comparison robust to short-term appearance changes (a detection only needs to look like the track at *some* recent frame, not the last one). The pair is rejected if this distance is above a threshold. The final association cost passed to the Hungarian algorithm is a weighted sum of the two terms.

Matching Cascade

DeepSORT does not solve a single global assignment over all tracks at once. It runs the Hungarian algorithm in several passes, ordered by the *age* of the track since its last successful match. Tracks that have just been observed are matched first, and tracks that have not been seen for several

frames only get a chance once the easier matches are taken. The motivation is that recently observed tracks have tighter Kalman uncertainties and more reliable appearance galleries, so giving them priority avoids stealing their detections in favour of an older, drifting track. A final IoU-based matching pass picks up tentative tracks and sudden appearance changes that the cascade might have left unmatched.

Appearance Network

DeepSORT itself is a tracking framework that combines motion cues with appearance-based data association. In its original implementation, the appearance term is computed using a specific 128-dimensional pedestrian Re-ID network, namely a Wide Residual Network with two convolutional layers, six residual blocks, and a final dense layer. This descriptor is pretrained offline on MARS, a large-scale person re-identification dataset.

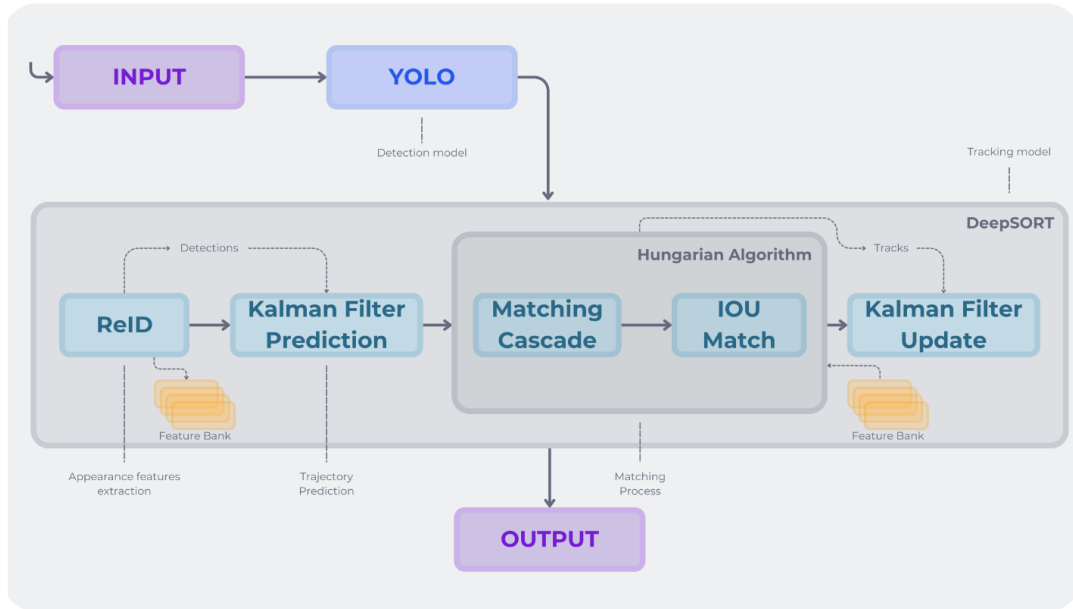


Figure 2.2: DeepSORT mechanism [1].

2.2.3 StrongSORT

StrongSORT [11] is a direct successor of DeepSORT. It keeps the Kalman state, the Hungarian association and the joint use of motion and appearance, but revisits three sub-components, the appearance branch, the noise model of the Kalman filter, and the handling of camera motion.

Stronger Appearance Embeddings and EMA Smoothing

The first change concerns the embeddings themselves. DeepSORT uses the small CNN, StrongSORT replaces it by a stronger person Re-ID backbone trained with the *Bag-of-Tricks* (BoT) [12]. BoT is a collection of training practices that together produce noticeably more discriminative person embeddings than the original DeepSORT network.

The second change concerns how those embeddings are stored. Where DeepSORT keeps a gallery of recent descriptors per track, StrongSORT stores a single appearance vector that is updated every frame by an *exponential moving average* (EMA). Let f_t be the embedding extracted at time t and $\bar{f}_{t-1}$ the stored track embedding from the previous frame, the update is

$$\bar{f}_t = \alpha \bar{f}_{t-1} + (1 - \alpha) f_t, \quad \alpha \in [0, 1).$$

Each new descriptor is averaged with a long history of past ones, so a single noisy frame (motion blur, partial occlusion, an extreme pose) only marginally shifts the track representation. The EMA does not, however, recover information that the underlying Re-ID model does not encode. It stabilises strong embeddings, but offers little when the embeddings themselves are weak or poorly adapted to the target domain.

Confidence-Aware Kalman Filter

DeepSORT runs a constant-velocity Kalman filter with fixed process and measurement noise. StrongSORT replaces it by a *Noise-Scale-Adaptive Kalman filter* (NSA-Kalman). The measurement noise of a detection is no longer constant but scales with the detector’s confidence score. Low-confidence detections are then treated as less reliable observations and induce larger noise, while high-confidence detections lead to tighter updates. Spurious or partially occluded detections therefore do not pull the Kalman state as much as they would in DeepSORT.

Camera Motion Compensation

DeepSORT either assumes a static camera or absorbs camera motion implicitly into the Kalman uncertainty. Even moderate handheld jitter or panning is enough to break this assumption, because the predicted track position then drifts together with the background instead of with the object. StrongSORT addresses this with an explicit *Camera Motion Compensation* (CMC) step based on *Enhanced Correlation Coefficient maximisation* (ECC) [13], an image-alignment algorithm that estimates the global geometric transformation between two consecutive frames by maximising the correlation of their pixel intensities. Predicted track states are then warped by this transformation so that both tracks and detections live in the coordinate system of the new frame, and the association cost only has to model object-relative motion.

Resulting Association

The final association cost in StrongSORT keeps the same shape as in DeepSORT (motion gating combined with appearance similarity), but the more discriminative embeddings, the smoother appearance state, and the noise-aware motion model jointly produce a more reliable cost matrix. The reported effect is fewer identity switches during long occlusions and fewer premature track terminations in crowded scenes.

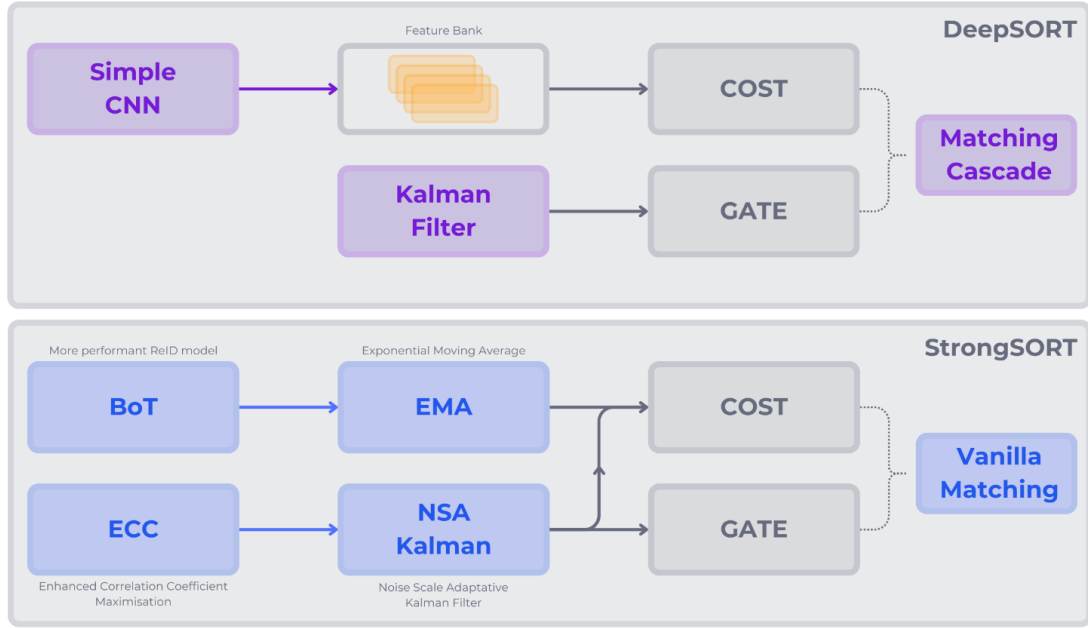


Figure 2.3: Comparison between the DeepSORT and StrongSORT frameworks [1].

2.2.4 ByteTrack

DeepSORT and StrongSORT both keep only the detections that are confidently above a single score threshold and discard everything else. ByteTrack [14] starts from a simple counter-observation: *a sizeable fraction of the discarded low-score boxes are real objects momentarily weakened by occlusion or motion blur, and dropping them is what causes most of the identity switches in crowded scenes.*

Lowering the threshold to recover them is not a solution either, because it floods the tracker with false positives that look just like genuine new objects. ByteTrack’s idea is to keep both groups of detections but to use them differently so that low-score boxes can extend existing tracks without being allowed to create new ones.

The BYTE Association Algorithm

At each frame, the detector output is split into two pools by score:

- **High-score detections** (score above a threshold τ_{high}): confident boxes that almost certainly correspond to real objects.
- **Low-score detections** (score between a small noise floor τ_{low} and τ_{high}): plausible boxes that include both occluded objects and false positives.

The Kalman filter predicts where each existing track should be in the new frame, and the matching then proceeds in two consecutive passes.

First pass: confident detections. The high-score detections are matched against every existing track using $1 - \text{IoU}$ as the cost, solved by the Hungarian algorithm. Pairs whose overlap is too small are rejected by a gating threshold. This is the regime in which a conventional tracker would already operate, and most associations are resolved here. Tracks that find a partner are

updated normally, and high-score detections that find no partner are kept aside for new-track creation.

Second pass: rescue with low-score detections. Tracks that the first pass left unmatched usually correspond to objects that the detector saw, but only weakly. Partial occlusion, motion blur, or an awkward pose can cause a detection to have a low score. A single-threshold tracker would terminate them. ByteTrack instead tries to match them against the low-score pool, again using only $1 - \text{IoU}$. The reasoning is geometric, a low-score box that lands precisely where the Kalman filter predicted an existing track is far more likely to be that track’s object than to be background noise, because background false positives have no reason to align with a predicted trajectory. Appearance is deliberately not used in this pass, because Re-ID embeddings extracted from low-confidence crops are themselves unreliable, while spatial overlap with a Kalman prediction is not. Low-score detections that find no track after this rescue pass are dropped as background.

New tracks. Only the high-score detections left over after the first pass are eligible to start a new track, and only if their score also exceeds an initialisation threshold τ_{init} (usually equal to or slightly above τ_{high}). This asymmetry is the safety net that makes the second pass safe. Low-score boxes can rescue an existing identity but can never create a new one, so the false positives they contain cannot pollute the set of tracks.

Track Lifecycle

A track is in one of three states. It is *tracked* when it received a detection in the current frame, *lost* when it has missed at least one frame but is still being predicted by the Kalman filter, and *removed* once it has been lost for more than a tolerance τ_{lost} frames. Lost tracks remain visible to the first matching pass throughout this grace period, so an individual that disappears behind an obstacle for a couple of seconds and reappears with a confident detection is reattached to its original identity rather than starting a new one. The grace period and the two-pool split are independent mechanisms, the first decides how long a track can survive without observations, the second decides how the observations are consumed.

Motion Model and the Role of Re-ID

The motion model itself is unchanged from SORT and DeepSORT explained above. A constant-velocity Kalman filter on the bounding-box centre, aspect ratio and height, with no camera-motion compensation. An appearance term (cosine distance between Re-ID embeddings) can optionally be added to the cost matrix of the first pass when a Re-ID network is available, but the published evaluation shows that this addition brings only a marginal gain on the tracking benchmark *MOT17* discussed in the following subsection (see section 2.2.6). This is consistent with the structure of the algorithm, the difficult associations that produce identity switches mostly happen in the second pass, where appearance is not used at all.

2.2.5 BoT-SORT

BoT-SORT [15] is a tracking-by-detection method that builds directly on ByteTrack’s two-stage association and augments it with three improvements to make the motion model and the appearance gating more robust. It was introduced as a *bag of tricks* (collection of techniques and improvements) rather than a redesign. ByteTrack’s overall pipeline (Kalman prediction, two-stage matching, track lifecycle) is preserved, and the additions can be enabled or disabled independently. The variant that includes the appearance branch is referred to as *BoT-SORT-ReID*, the variant without it as *BoT-SORT*.

A Direct Width–Height Kalman State

DeepSORT and most of its successors parameterise the bounding box through its centre, its aspect ratio, and its height (x_c, y_c, a, h) , leaving the width to be reconstructed as $a \cdot h$. Aharon et al. observe that this choice systematically degrades the width estimate in long predictions, because the aspect-ratio velocity has no direct geometric meaning and tends to over or under-shoot. BoT-SORT replaces this with a state vector that estimates width and height directly:

$$\mathbf{x} = (x_c, y_c, w, h, \dot{x}_c, \dot{y}_c, \dot{w}, \dot{h})^\top,$$

together with the measurement vector $\mathbf{z} = (z_{x_c}, z_{y_c}, z_w, z_h)^\top$. The process and measurement noise covariance matrices Q_k and R_k are scaled with the predicted width and height so that the filter behaves consistently across object sizes. This modification is reported to improve the metric HOTA (introduced later in section 2.4.4) by producing predicted boxes whose width tracks the detector output more faithfully and therefore aligns better in the IoU-based association step, as shown in fig. 2.4.

Camera-Motion Compensation (CMC)

For the same reason that motivates the ECC stabilisation of StrongSORT, BoT-SORT compensates for camera motion explicitly through a *global motion compensation* (GMC) step. The fig. 2.5 visually represents these three steps:

1. Sparse keypoints are extracted from frame f_{k-1} and tracked into frame f_k using *pyramidal Lucas–Kanade* sparse optical flow, the algorithm that estimates a per-keypoint displacement under brightness-constancy and small-motion assumptions. It is robust to large inter-frame displacements as it is running on a Gaussian image pyramid from coarse to fine resolution. Only the keypoints with consistent forward–backward tracking are retained and they provide a sparse set of pixel correspondences between the two frames.
2. A 2×3 affine warp $A_{k-1}^k \in \mathbb{R}^{2 \times 3}$ is then fitted to these correspondences with the *Random Sample Consensus* (RANSAC) algorithm [16]. RANSAC is a robust estimator for model fitting in the presence of outliers:
 - It repeatedly samples a small random subset of correspondences (here three pairs, the minimum needed to determine an affine transform).
 - It fits a candidate warp to that subset.

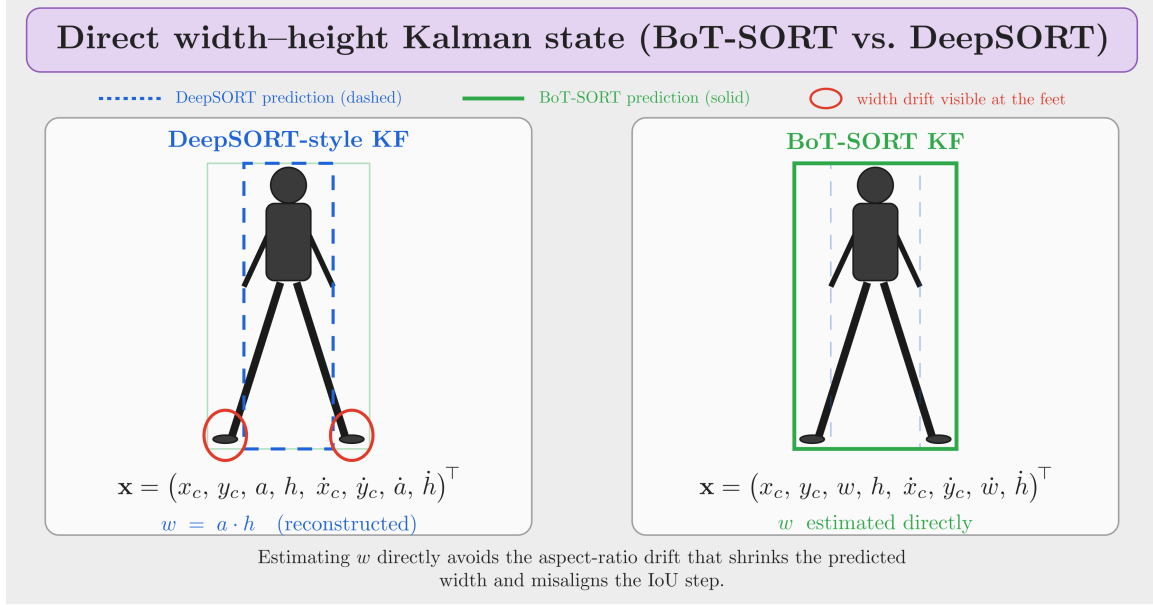


Figure 2.4: Effect of BoT-SORT’s direct width–height Kalman state on the predicted bounding box. **Left:** the DeepSORT-style state reconstructs the width as $w = a \cdot h$. The aspect-ratio velocity $\dot{a}$ carries no direct geometric meaning, so over long predictions the reconstructed width drifts (dashed blue), shrinks toward the torso, and intersects the legs of the walking pedestrian (red ovals). **Right:** BoT-SORT replaces the state and scales the process with the predicted width and height. The width is now estimated directly, and the predicted box (solid green) hugs the silhouette tightly.

- It counts how many of the remaining correspondences agree with this warp up to a pixel-distance.
- It returns, after a fixed number of iterations, the warp with the largest inlier set tolerance (the *inliers*).

In the BoT-SORT pipeline this is what isolates the *rigid background motion* from the non-rigid motion of the chimpanzees themselves. Keypoints that fall on a moving animal are inconsistent with the dominant warp and end up as outliers, so the estimated A_{k-1}^k describes the camera motion alone.

3. Before association, every predicted Kalman state and its covariance matrix are warped according to A_{k-1}^k . This mechanism re-expresses the tracks in the coordinate system of the new frame.

The magnitude of the warp depends on how much the camera itself moves. In strictly static-camera footage the affine matrix collapses to the identity and the step has no effect, whereas in handheld or panning footage it can shift the predicted Kalman state by tens of pixels per frame. CMC is therefore a strict generalisation of the static-camera Kalman filter rather than a trade-off.

IoU–Re-ID Fusion

The third modification concerns how appearance information is combined with motion. DeepSORT and StrongSORT compute the cost matrix as a weighted sum of an IoU distance and a

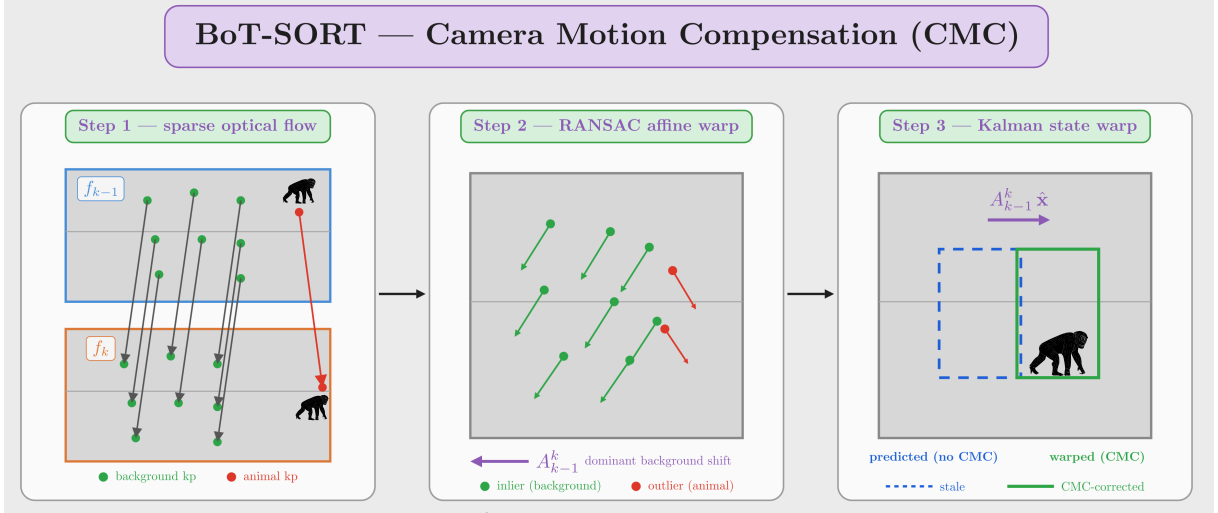


Figure 2.5: The three-step Camera Motion Compensation (CMC) pipeline of BoT-SORT. **Step 1 (sparse optical flow)**. Sparse keypoints are extracted from frame f_{k-1} and tracked into frame f_k . Green keypoints lie on the static background while the red keypoint lies on a moving animal and follows a different displacement vector. **Step 2 (RANSAC affine warp)**. A 2×3 affine warp A_{k-1}^k is robustly fitted to the correspondences. RANSAC identifies the dominant background shift (inliers) and rejects keypoints on the animal (outliers), so the estimated warp describes camera motion alone. **Step 3 (Kalman state correction)**. Before association, every predicted Kalman state is re-expressed in the coordinate system of frame f_k by applying $A_{k-1}^k \hat{\mathbf{x}}$. The dashed blue box shows the stale prediction (camera motion unaccounted for). The solid green box shows the CMC-corrected prediction that now overlaps the true silhouette.

cosine distance between Re-ID embeddings. BoT-SORT argues that this linear combination is fragile. It forces every association to trust both signals simultaneously, and this mechanism often fails when one of them is uninformative. Instead, the authors define the appearance cost as a *gated* cosine distance,

$$\hat{d}_{i,j}^{\cos} = \begin{cases} 0.5 \cdot d_{i,j}^{\cos} & \text{if } d_{i,j}^{\cos} < \theta_{\text{emb}} \text{ and } d_{i,j}^{\text{IoU}} < \theta_{\text{IoU}}, \\ 1 & \text{otherwise,} \end{cases}$$

and combine the two distances by taking their element-wise minimum:

$$C_{i,j} = \min(d_{i,j}^{\text{IoU}}, \hat{d}_{i,j}^{\cos}).$$

Two thresholds parameterise the gate. The *proximity threshold* θ_{IoU} discards spatially implausible matches before any appearance comparison takes place. The *appearance threshold* θ_{emb} sets the minimum embedding similarity required for the appearance branch to contribute. Embeddings are extracted only for high-confidence detections, and the per-track appearance state is updated frame-by-frame by an exponential moving average, in the same spirit as StrongSORT. This design lets either modality unilaterally validate a match while requiring both to agree before a borderline association is accepted, and it makes the appearance and proximity gates two interpretable hyperparameters rather than a single opaque fusion weight.

Positioning Relative to ByteTrack and StrongSORT

BoT-SORT can be read as a controlled extension of ByteTrack. It preserves the two-stage association on high and low-confidence detections from ByteTrack, and the three additions touch only the motion model, the predicted-box geometry, and the cost matrix used in the first stage. Compared to StrongSORT, BoT-SORT shares the use of stronger Re-ID embeddings and an EMA appearance state, but its motion correction is global (frame-level affine warp) rather than per-track NSA-Kalman, and its appearance–motion fusion is gate-and-minimum rather than a linear combination.

2.2.6 Comparison on MOT17

Figure 2.6 and Table 2.1 report the published MOT17 results of the four trackers above, together with the original SORT baseline. MOT17 is the standard pedestrian benchmark used to compare tracking methods under controlled conditions. We restrict the comparison here to this small family of trackers, since broader evaluations can be found in the BoT-SORT and StrongSORT papers themselves. The two reported metrics, *MOTA* and *HOTA*, are defined later in sections 2.4.2 and 2.4.4. At this stage, it is sufficient to read them as “higher is better”. The three modern trackers (StrongSORT, ByteTrack, BoT-SORT) lie within roughly one HOTA point of each other on this benchmark, while DeepSORT and SORT trail by a wider margin.

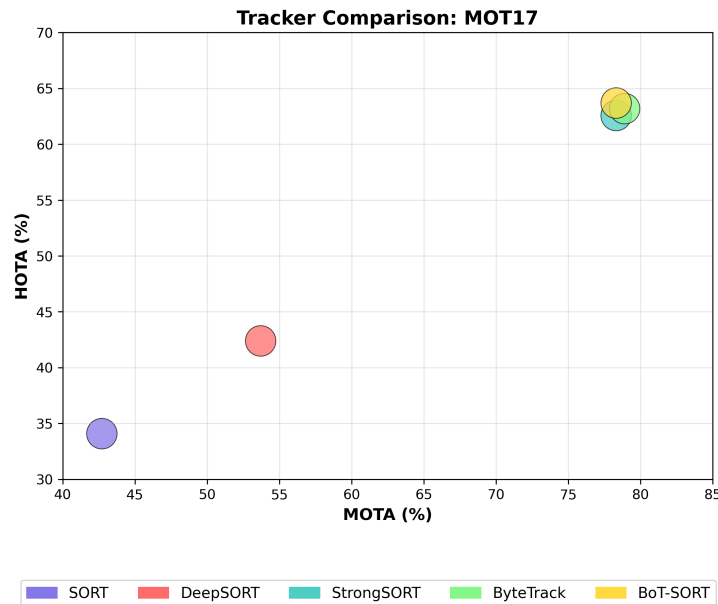


Figure 2.6: MOTA-HOTA performance comparison of five SORT-family trackers evaluated on the MOT17 benchmark

Table 2.1: Comparison of SORT-family trackers on MOT17. MOTA and HOTA are defined later in sections 2.4.2 and 2.4.4.

Tracker	MOTA $\uparrow$	HOTA $\uparrow$
SORT	42.7	34.1
DeepSORT	53.7	42.4
StrongSORT	78.3	62.6
ByteTrack	78.9	63.2
BoT-SORT	78.3	63.7

2.3 Re-ID models

Re-Identification refers to the task of deciding whether two image crops, taken at different times or from different viewpoints, depict the same individual. A Re-ID model is trained to map each crop to a fixed-dimensional vector, called an *embedding*, such that two crops of the same individual produce nearby vectors and two crops of different individuals produce distant ones. The notion of distance is usually the cosine distance $1 - \cos(\mathbf{u}, \mathbf{v})$, which only depends on the angle between the vectors and is therefore invariant to their norm. Re-ID embeddings are used as the appearance term in the association cost of trackers such as DeepSORT, StrongSORT, and BoT-SORT.

2.3.1 OSNet: Omni-Scale Feature Learning

OSNet [17] (*Omni-Scale Network*) is a convolutional Re-ID backbone originally trained on the pedestrian benchmark *Market-1501*. It is widely used as the appearance encoder in tracking-by-detection pipelines: each detected box is cropped from the frame, resized, and passed through OSNet to obtain a vector compared to the per-track embeddings stored from previous frames.

The Multi-Scale Problem in Re-ID

Re-ID is hard because two different sources of variability act simultaneously. The same individual can look very different across frames because of viewpoint, pose, partial occlusion, or lighting (*intra-identity variation*), while two different individuals may share a similar global silhouette or clothing colour (*inter-identity variation*). A discriminative embedding therefore has to combine information at several spatial resolutions. The overall silhouette and colour distribution, fine local cues such as a logo or a texture pattern, and the relation between a local detail and its broader spatial context. A network that only sees one scale at a time tends either to confuse two individuals with similar global appearance, or to miss the local cues that separate them.

Core Architecture

OSNet builds the embedding through a stack of *multi-stream residual blocks*. Each block runs the input feature map through several convolutional streams in parallel. The streams differ in the size of their receptive field, so shallow streams capture fine local details while deeper streams capture larger spatial context. Their outputs are then merged into a single feature map by an *aggregation gate*, a small subnetwork that, for every channel, predicts an input-dependent weight

applied to each stream. The same image can therefore be represented by a different mix of scales depending on which scale carries the most information for that particular crop. The final embedding is obtained by global average pooling of the last block’s output.

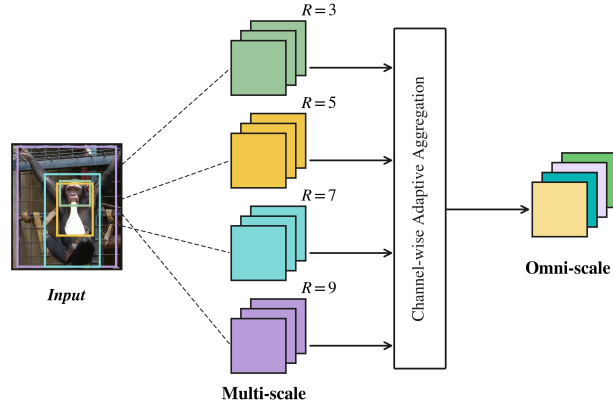


Figure 2.7: Schematic illustration adapted from [17] of an OSNet omni-scale residual block. The input feature map is processed in parallel through convolutional streams with different receptive-field sizes ($R = 3, 5, 7, 9$), and a channel-wise aggregation gate fuses them into a single omni-scale representation.

OSNet is released in several width variants (`osnet_x0_25`, `osnet_x0_5`, `osnet_x0_75`, `osnet_x1_0`) trading parameter count for embedding quality. All public checkpoints are trained on person Re-ID datasets (*Market-1501*, *DukeMTMC-reID*, *MSMT17*), so the embeddings are tuned to discriminate pedestrians.

2.3.2 ChimpUFE

ChimpUFE [3] (*Universal Chimpanzee Face Embedder*) is a Re-ID model trained on chimpanzee faces. Unlike OSNet, which encodes a body crop, ChimpUFE encodes a face crop, and unlike the supervised face-recognition models on which earlier chimpanzee identification systems were built, it is trained without identity labels. Its design target is *open-set* chimpanzee identification. Given a query face, retrieve other faces of the same individual from a gallery, and refuse to match when the individual is absent. Open-set is the realistic regime for wildlife monitoring, where new individuals appear over time and the closed list of training identities cannot be assumed to cover them.

Why Not Train a Supervised Face Classifier

The straightforward approach to chimpanzee face identification is to take a supervised face-recognition loss such as *ArcFace* [18] or *CosFace* [19] and train a classifier over a fixed set of named individuals. Two limitations make this impractical at population scale. First, those losses need a large number of labelled images per identity, and identity labels for wild chimpanzees require expert annotators and have to be redone for every new study site. Second, a closed-set classifier cannot, by construction, decide that a query face does not belong to any of the training identities, which is the most common situation when monitoring an unknown population.

ChimpUFE replaces identity supervision by self-supervised pretraining on a much larger pool of unlabelled face crops. The trained model is then used for retrieval. At inference time, a query embedding is compared by cosine similarity to a gallery of stored embeddings, and the nearest neighbour determines the outcome. No classification head is involved.

Self-Supervised Backbone

The backbone of ChimpUFE is a *Vision Transformer* (ViT), a network that splits the input image into a grid of small patches, embeds each patch as a token, and processes the sequence with self-attention layers. Two variants are released, with 22M and 87M parameters respectively, the smaller one targeting inference speed and the larger one targeting embedding quality.

The backbone is trained with the *DINOv2* [20] framework. DINOv2 is a self-distillation method. It works with two networks with identical architecture. A *student* and a *teacher*, are fed with two different augmented crops of the same image, and the student is trained so that its output distribution matches the teacher’s. The teacher is not trained directly, its weights are an exponential moving average of the student’s. No identity labels and no explicit negative pairs are used. DINOv2 embeddings are reported to cluster well under cosine distance without further fine-tuning, which is the property required by nearest-neighbour retrieval.

Mining Training Faces Without Identity Labels

Self-supervised pretraining only removes the need for *identity* labels, the model still needs a large pool of face crops to learn from. ChimpUFE constructs this pool automatically from raw camera-trap videos through a three-stage pipeline. A *YOLOX* face detector first locates candidate faces in every frame. ByteTrack is then used to link those detections across frames, which both recovers faces missed by the detector during short occlusions and discards isolated false positives that fail to form a track. Detections inside the surviving tracks are finally ranked by detector confidence and subsampled so that only sharp, well-framed crops feed the training set. Applied to several camera-trap corpora, this pipeline yielded roughly 0.6 million face crops without any manual identity labelling.

Inference Protocol

ChimpUFE is not a tracker. It produces a single embedding per face crop, and assumes that an upstream pipeline has already extracted those crops, optionally grouped into short tracklets by a face tracker. Identity assignment then reduces to nearest-neighbour search in embedding space, performed either per crop or per tracklet, against a gallery of labelled embeddings.

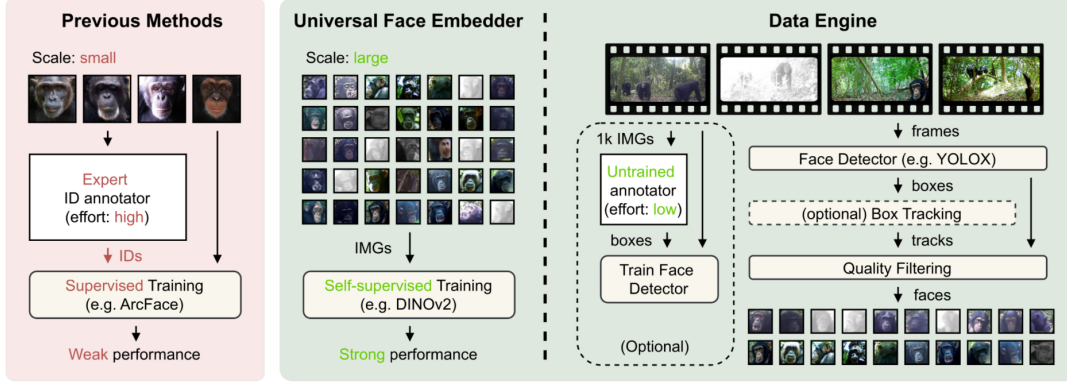


Figure 2.8: Overview of the pipeline used to learn universal chimpanzee face embeddings [3]. **Left:** prior approaches rely on expert identity annotation and supervised losses such as ArcFace, which limits scalability in open-set scenarios. **Middle:** the proposed approach trains a face embedding on unlabelled camera-trap data with DINOv2, producing a retrieval-oriented representation. **Right:** the data-extraction pipeline that mines high-quality face crops from raw videos using a lightweight face detector, a tracker, and confidence-based filtering.

2.3.3 ArcFace: Angular Margin Loss for Face Recognition

ArcFace [18] is a loss function designed for face recognition that operates directly in the embedding space. All embeddings and class prototypes are ℓ_2 -normalised, so they live on a hypersphere and their pairwise similarities reduce to cosines of angles. An *angular margin* m is added to the angle θ_{i,y_i} between a sample’s embedding and *its own* class prototype before computing the softmax loss. This acts like a penalty during training. It forces embeddings of the same individual to cluster more tightly around their prototype (intra-class compactness), and it widens the angular gap between distinct classes (inter-class separability). Because each class is now a tight cluster, gaps appear naturally between different identities on the hypersphere.

Formally, for sample i with ground-truth class y_i , the logit assigned to class k is:

$$\ell_{i,k} = s \cdot \begin{cases} \cos(\theta_{i,k} + m) & \text{if } k = y_i, \\ \cos(\theta_{i,k}) & \text{otherwise,} \end{cases} \quad (2.1)$$

where s is a scale factor that controls the magnitude of the logits before the softmax. Adding m to θ_{i,y_i} *decreases* the cosine, and therefore the logit, of the correct class. The model is only credited for a correct prediction once the sample’s embedding lies sufficiently close to its prototype to compensate for the extra margin. Unlike contrastive or triplet losses, ArcFace requires no explicit pair or triplet mining, the margin acts implicitly on every batch through the softmax. Although ArcFace is trained in a closed-set classification setting, its normalised embeddings are commonly used for retrieval by cosine similarity. Open-set operation is then implemented by applying a similarity threshold and rejecting queries whose nearest neighbour is not sufficiently close. The quality of this rejection depends on threshold calibration and on the similarity between the deployment domain and the training data.

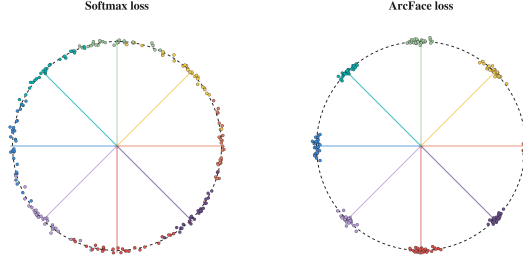


Figure 2.9: Geometric comparison of softmax and ArcFace embedding spaces

2.4 Evaluation Metrics for MOT

Evaluating multi-object tracking is inherently difficult because the task simultaneously requires accurate detection, precise localisation, and consistent identity association over time. No single scalar can capture all three dimensions without making trade-offs, and the choice of metric directly influences how researchers tune their systems.

This section first introduces the standard metrics for Re-Identification (Re-ID) evaluation, Cumulative Matching Characteristics (CMC) and mean Average Precision (mAP) which are used to assess appearance models independently of the tracker. We then review the three main families of MOT metrics, MOTA, IDF1, and HOTA and explain why we adopt HOTA as the primary tracking evaluation metric throughout this thesis.

2.4.1 Re-Identification Metrics: CMC and mAP

Before evaluating the full tracking pipeline, it is necessary to assess the quality of appearance embeddings produced by Re-ID models such as OSNet [17]. The standard evaluation protocol, formalised by Zheng et al. [21], uses a *query/gallery* paradigm, for each query image of an individual, the model ranks all gallery images by embedding distance, and two complementary metrics measure retrieval quality.

Cumulative Matching Characteristics (CMC)

The CMC curve measures the probability that the correct identity appears within the top- k ranked gallery images. The most commonly reported value is **Rank-1 accuracy**: the fraction of queries for which the closest gallery image (by embedding distance) belongs to the same identity as the query.

$$\text{Rank-}k = \frac{1}{|Q|} \sum_{q \in Q} \mathbf{1}[\text{correct match within top-}k \text{ of gallery}] \quad (2.2)$$

Where Q is the set of query images and $\mathbf{1}[\cdot]$ is the indicator function. Rank-1 captures whether the model’s single best guess is correct, making it an intuitive measure of discriminative power. However, CMC has a limitation as it only credits the *first* correct match and ignores whether additional correct gallery images (e.g., other views of the same individual) are also highly ranked.

Mean Average Precision (mAP)

Mean Average Precision (mAP) [21] addresses this limitation by evaluating the full ranked list. For a query q with n_q correct gallery matches, the Average Precision is:

$$\text{AP}(q) = \frac{1}{n_q} \sum_{k=1}^G \text{Precision}(k) \cdot \mathbf{1}[\text{item at rank } k \text{ is correct}] \quad (2.3)$$

Where G is the total gallery size and $\text{Precision}(k)$ is the fraction of correct matches among the top- k results. The mAP is the mean of AP over all queries:

$$\text{mAP} = \frac{1}{|Q|} \sum_{q \in Q} \text{AP}(q) \quad (2.4)$$

mAP rewards models that rank *all* correct gallery images highly, not just the single nearest one. It is therefore more informative than Rank-1 when multiple views of each identity exist in the gallery which is the case in our chimpanzee dataset.

2.4.2 MOTA

The CLEAR MOT metrics [22] introduced MOTA (Multi-Object Tracking Accuracy) as the standard for tracking evaluation. MOTA operates at the *detection level* which means that in each frame a bijective (one-to-one) matching is computed between ground-truth detections and predicted detections using the Hungarian algorithm. In addition, the matches are accepted only if their spatial overlap meets a minimum threshold α . Matched pairs become true positives (TP), unmatched ground-truth detections are false negatives (FN), unmatched predictions are false positives (FP). Association is measured through *identity switches* (IDSW). A TP is counted as an IDSW when its predicted identity differs from that of the previous TP belonging to the same ground-truth trajectory.

$$\text{MOTA} = 1 - \frac{|\text{FN}| + |\text{FP}| + |\text{IDSW}|}{|\text{gtDet}|} \quad (2.5)$$

MOTA suffers from several well-documented biases. First, detection errors (FN + FP) typically outnumber identity switches by two orders of magnitude (on the MOT17 benchmark the ratio $(|\text{FN}| + |\text{FP}|) : |\text{IDSW}|$ averages 98.6 across top-performing trackers [2]). Consequently, MOTA functions almost exclusively as a detection metric, a linear regression between MOTA and the detection-only MODA score yields $R^2 = 0.994$, whereas the correlation with IDSWs is only $R^2 = 0.237$. Second, MOTA is asymmetric between precision and recall (precision errors are penalised hyperbolically while recall errors are penalised linearly) which incentivises trackers to favour high precision at the expense of recall. Third, association is measured only against the *single previous* detection (first-order), so MOTA cannot evaluate long-term trajectory consistency and actually penalises trackers that correct their own identity mistakes. Finally, MOTA scores are unbounded below (they can be negative) which makes them difficult to interpret.

2.4.3 IDF1

IDF1 [23] was designed to address MOTA’s neglect of association by shifting evaluation to the trajectory level. A bijective mapping is computed between the sets of ground-truth trajectories and predicted trajectories. Overlapping segments of matched trajectory pairs become identity true positives (IDTP), non-overlapping segments and unmatched trajectories become identity false negatives (IDFN) and identity false positives (IDFP). The IDF1 score is the harmonic mean of identity recall and identity precision:

$$\text{IDF1} = \frac{|\text{IDTP}|}{|\text{IDTP}| + 0.5 |\text{IDFN}| + 0.5 |\text{IDFP}|} \quad (2.6)$$

While IDF1 better captures association quality than MOTA, it introduces the opposite bias as it is non-monotonic with respect to detection. Because only the single best matching trajectory contributes positively, additional correct detections in non-matched trajectory segments actually *decrease* the IDF1 score. This counter-intuitive property has prevented IDF1 from being adopted as the primary metric by any single-camera MOT benchmark. Furthermore, IDF1 ignores association quality outside matched trajectory sections and does not account for localisation accuracy.

2.4.4 HOTA

HOTA (Higher Order Tracking Accuracy) [2] was proposed as a unified metric that explicitly and equally balances detection accuracy and association accuracy, while also incorporating localisation quality. It is the primary evaluation metric used throughout this thesis.

Matching and True Positive Associations

Like MOTA, HOTA performs matching at the *detection level*. This yields sets of TPs, FNs, and FPs. HOTA introduces three quantities, *True Positive Associations* (TPA), *False Negative Associations* (FNA), and *False Positive Associations* (FPA). These quantities measure how well the trajectories passing through each TP align globally over time. For a given TP c :

- TPA(c): TPs sharing both the same ground-truth ID and the same predicted ID as c .
- FNA(c): ground-truth detections with the same ground-truth ID as c but assigned a different (or no) predicted ID. These measure *track fragmentation*.
- FPA(c): predicted detections with the same predicted ID as c but assigned a different (or no) ground-truth ID. These measure *identity contamination* (mergers).

Detection Accuracy (DetA)

Detection Accuracy measures the alignment between the set of all ground-truth and predicted detections using the Jaccard index:

$$\text{DetA}_\alpha = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}| + |\text{FP}|} \quad (2.7)$$

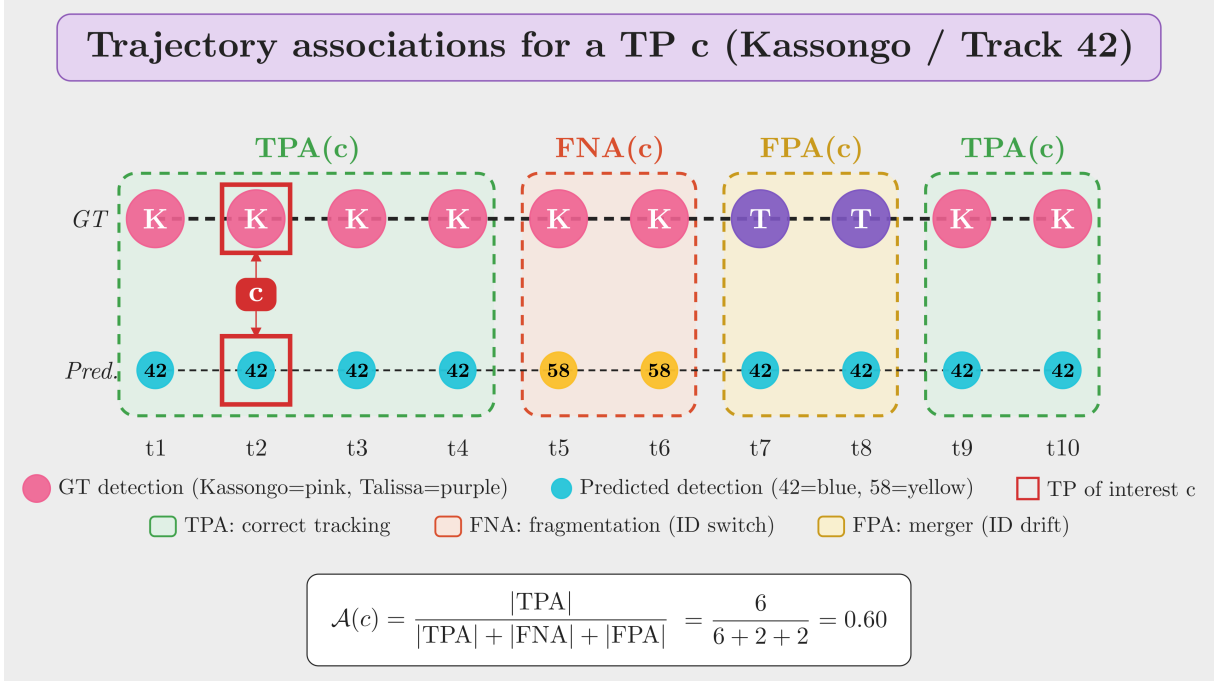


Figure 2.10: Illustration of TPA, FNA and FPA adapted to our chimpanzee tracking setting. The ground-truth trajectory of Kassongo is tracked by **Track 42**. The TP of interest c occurs at frame t_2 . **TPA**(c) groups all frames where **Track 42** correctly follows Kassongo. **FNA**(c) highlights frames where Kassongo is present but the tracker has switched to **Track 58** after an occlusion (the fragmentation error that drives AssRe down). **FPA**(c) highlights frames where **Track 42** has drifted onto Talissa (the merger error that drives AssPr down). The association score for this TP is 0.60.

DetA decomposes into Detection Recall (DetRe) and Detection Precision (DetPr). They measure respectively the fraction of ground-truth detections that are correctly predicted and the fraction of predictions that correspond to a real object. Unlike MOTA, DetA treats recall and precision symmetrically through the Jaccard formulation.

Association Accuracy (AssA)

Association Accuracy is the metric most directly relevant to the core challenge of our pipeline (maintaining consistent identity assignment across long video sequences). For each TP c , an association score $\mathcal{A}(c)$ is computed as the Jaccard index over the trajectory pair passing through c :

$$\mathcal{A}(c) = \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FNA}(c)| + |\text{FPA}(c)|} \quad (2.8)$$

The overall AssA is the average of $\mathcal{A}(c)$ over all TPs:

$$\text{AssA}_\alpha = \frac{1}{|\text{TP}|} \sum_{c \in \{\text{TP}\}} \mathcal{A}(c) \quad (2.9)$$

This formulation evaluates association *globally* over the entire video. Each TP’s association score reflects how well its predicted trajectory aligns with its ground-truth trajectory across all frames, not just the single previous frame as in MOTA. This is what makes HOTA a *higher-order* metric. It captures long-term trajectory coherence rather than frame-to-frame identity consistency. AssA further decomposes into two sub-metrics:

- **Association Recall (AssRe):** measures how well predicted trajectories cover ground-truth trajectories. A low AssRe indicates that the tracker fragments single individuals into multiple tracklets.
- **Association Precision (AssPr):** measures how well predicted trajectories stay within a single ground-truth trajectory. A low AssPr indicates that the tracker merges multiple individuals into a single tracklet.

$$\text{AssRe}_\alpha = \frac{1}{|\text{TP}|} \sum_{c \in \{\text{TP}\}} \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FNA}(c)|} \quad (2.10)$$

$$\text{AssPr}_\alpha = \frac{1}{|\text{TP}|} \sum_{c \in \{\text{TP}\}} \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FPA}(c)|} \quad (2.11)$$

This decomposition is a powerful diagnostic tool as a tracker that scores well on DetA but poorly on AssA can be diagnosed as suffering from association failures, while the further split between AssRe and AssPr distinguishes *fragmentation* (a single ground-truth trajectory split into many short tracklets, low AssRe) from *contamination* (a single predicted trajectory covering several individuals, low AssPr).

Localisation Accuracy (LocA)

HOTA incorporates localisation quality by computing the score at multiple spatial overlap thresholds $\alpha \in \{0.05, 0.10, \dots, 0.95\}$ and averaging. This makes the final HOTA score sensitive to how precisely bounding boxes are aligned, unlike MOTA and IDF1 which use a fixed α and ignore the degree of spatial overlap beyond the threshold.

$$\text{LocA} = \int_0^1 \frac{1}{|\text{TP}_\alpha|} \sum_{c \in \{\text{TP}_\alpha\}} \mathcal{S}(c) \, d\alpha \quad (2.12)$$

Where $\mathcal{S}(c)$ is the spatial similarity (IoU) between the ground-truth and predicted detections forming the TP c .

The HOTA Score

The final HOTA score at a given threshold α is defined as the geometric mean of DetA and AssA:

$$\text{HOTA}_\alpha = \sqrt{\text{DetA}_\alpha \cdot \text{AssA}_\alpha} \quad (2.13)$$

The overall HOTA score integrates over localisation thresholds:

$$\text{HOTA} = \int_0^1 \text{HOTA}_\alpha \, d\alpha \approx \frac{1}{19} \sum_{\alpha \in \{0.05, 0.10, \dots, 0.95\}} \text{HOTA}_\alpha \quad (2.14)$$

The geometric mean ensures that both detection and association contribute equally. A tracker that excels at detection but fails at association (or vice versa) cannot achieve a high HOTA score. The overall HOTA score is obtained by averaging HOTA_α over the range of localisation thresholds, thereby also incorporating localisation quality.

Properties of HOTA

Several properties make HOTA particularly well-suited for our evaluation:

1. **Balanced:** HOTA is the geometric mean of DetA and AssA, ensuring equal sensitivity to detection and association errors.
2. **Monotonic:** Improving any single error type (detection recall, detection precision, association recall, association precision, or localisation) strictly improves the HOTA score. Neither MOTA nor IDF1 possess this property.
3. **Higher-order association:** Unlike MOTA’s first-order IDSWs, HOTA evaluates trajectory alignment over the full video, rewarding trackers that maintain (or recover) correct identities over long time horizons.
4. **Error type differentiable:** The decomposition into DetRe, DetPr, AssRe, AssPr, and LocA enables precise diagnosis of where a tracker fails. This is essential for the iterative development of our pipeline.
5. **Bounded and interpretable:** HOTA scores lie in $[0, 1]$, unlike MOTA which can be negative.

A large-scale user study by Luiten et al. confirmed that HOTA aligns more closely with human

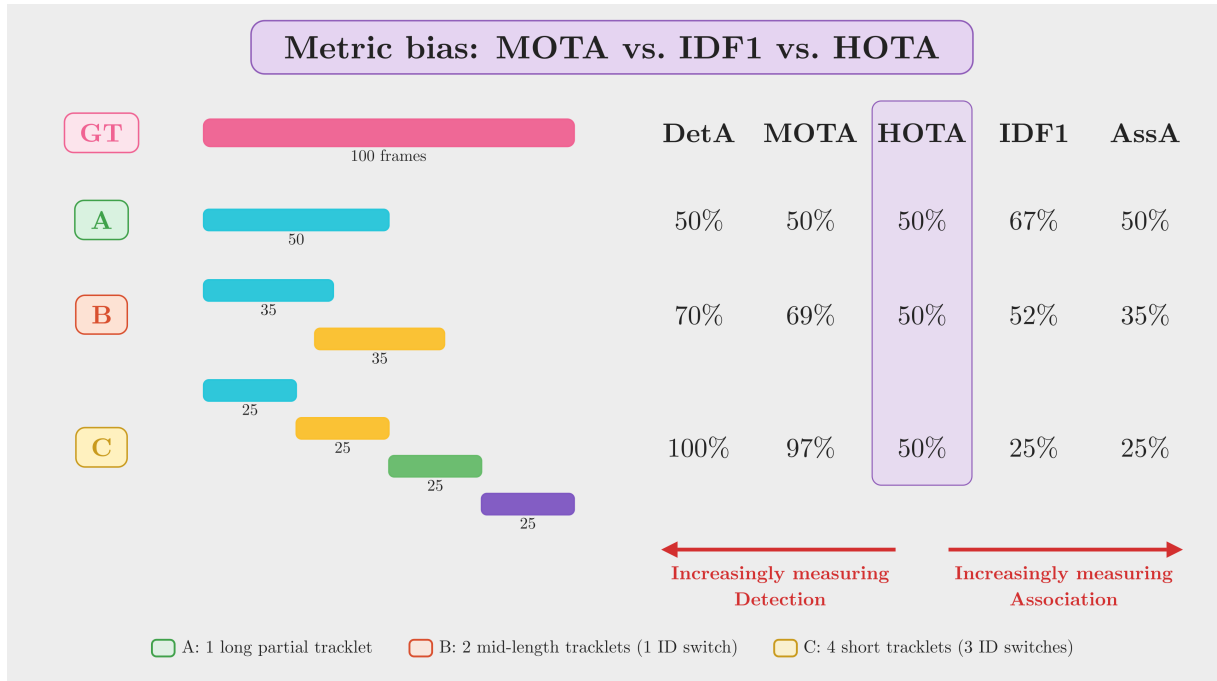


Figure 2.11: Illustration of metric biases (reproduced from [2]). Three trackers are shown in order of increasing detection accuracy and decreasing association accuracy.

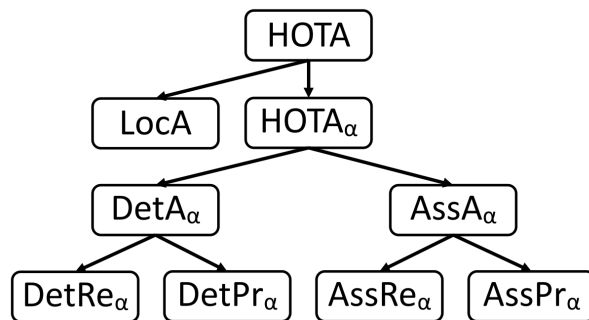


Figure 2.12: Decomposition of HOTA into sub-metrics [2]. HOTA decomposes into DetA and AssA, which further decompose into recall and precision components, enabling fine-grained diagnosis of tracking errors.

visual judgment of tracking quality than both MOTA and IDF1. Among MOT researchers, HOTA agreed with human rankings 80% of the time against MOTA and 86% against IDF1 [2].

2.4.5 Choice of Primary Metric

For the chimpanzee tracking pipeline evaluated in our thesis, HOTA is the natural choice as primary metric for two reasons. First, its symmetric treatment of detection and association removes the bias of MOTA towards detection and the bias of IDF1 towards association, both of which would obscure the trade-offs that arise when comparing online trackers, fine-tuned Re-ID models, and post-processing variants on the same footage. Second, the DetA/AssA/LocA decomposition and the further split of AssA into AssRe and AssPr provides the diagnostic granularity needed to identify which sub-problem dominates a given configuration and to design targeted interventions accordingly.

All experimental results in this thesis therefore report HOTA together with DetA and AssA. AssRe and AssPr are reported in addition when a diagnostic analysis of the failure mode is required.

2.5 The ChimpRec Pipeline: Starting Point

This thesis is a direct continuation of the ChimpRec system introduced by Cousin and De-meure [1] during the 2024–2025 academic year. Rather than redesigning the pipeline from scratch, our work inherits its modular structure and concentrates on the components that were identified as weak or incomplete. After reviewing the broader literature in the preceding sections, we summarise the architecture, the reported performance of each stage, and the limitations of the legacy ChimpRec pipeline that directly motivate our contributions.

2.5.1 Architecture

Tracking. The prior pipeline follows a tracking-by-detection design organised in two levels. At the *per-frame* level, three independent modules operate in a cascade on every frame. A YOLOv8s **body detector** locates all visible chimpanzees, a second YOLOv8s **face detector** is applied on each body crop, and an **individual-recognition** module attempts to assign a name to each detected face. At the *video* level, a multi-object tracker links the per-frame body detections into temporally coherent *chains of detections* (CODs), so that identity can in principle be assigned once per trajectory rather than once per frame. A manual correction interface sits at the output of the pipeline and allows an annotator to merge fragmented CODs, fix identity switches, and assign names, producing the clean ground-truth files on which evaluation relies.

Face recognition. The recognition branch of last year was designed as a closed-set face classifier over 20 known chimpanzees of the study group. In their work they use an InceptionResNetV1 backbone [24] pre-trained on VGGFace2 [25] and fine-tuned on the available chimpanzee face dataset detailed in the next chapter in section 3.5.3. Two fine-tuning strategies were tested

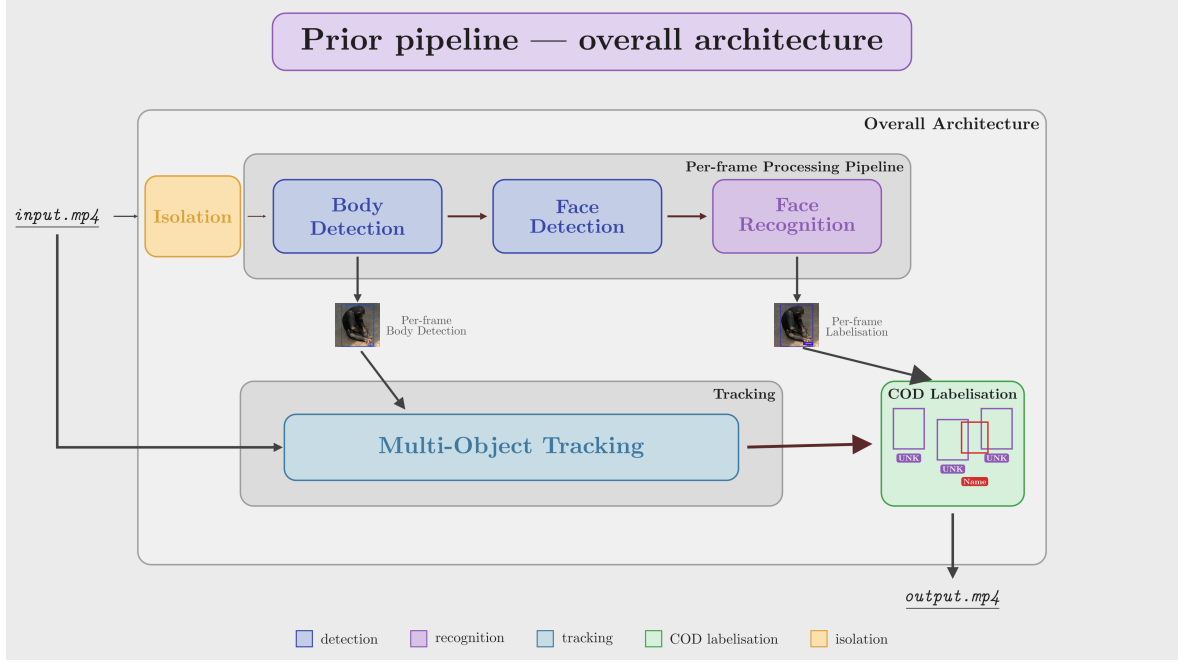


Figure 2.13: Overall architecture of the prior pipeline.

for training the model. A metric-learning approach based on triplet loss with k -NN classification, and a classification approach obtained by adding a fully connected layer on top of the backbone [1].

2.5.2 Components and Reported Performance

The four components of the prior pipeline reach uneven levels of maturity.

- **Body detection.** A YOLOv8s model trained on approximately 5 500 manually annotated frames, augmented with the public *Count*, *Crop and Recognise* and *ChimpBehave* datasets. On the test set it reaches 82% precision and 88% recall.
- **Face detection.** A YOLOv8s model trained on face crops obtained from the recognition dataset and public sources, reaching 84% precision and 79% recall. The dominant failure mode is confusion between the face and hairless regions of the body.
- **Individual recognition.** On their test set containing 132 images, these approaches reached 12.0% and 13.6% top-1 accuracy, respectively, with the best result obtained by the model trained with the fully connected layer. A separate proof-of-concept experiment trained the model with a larger chimpanzee dataset called CCR and reached 64.6% accuracy. They reported that the main bottleneck was not the architecture itself but the limited amount and diversity of labeled face data available for the chimpanzees.
- **Tracking.** DeepSORT is used with OSNet (osnet_x1_0) pre-trained on Market-1501 without any domain adaptation. The tracker is functional but exhibits frequent identity switches during occlusions and unjustified COD breaks when individuals briefly leave the frame or cross trajectories.

In the final integrated system, the recognition branch is unused. Its 13.6% accuracy is too

low to enrich the tracker’s association cost without introducing noise. The operational system therefore relies on body detection and DeepSORT alone, with the manual correction tool compensating for tracking errors. This is an important point for this thesis. The architecture evaluated in ChimpRecV1 [1] is a detector-plus-tracker pipeline supplemented by human corrections, not an end-to-end fully automated recognition system.

2.5.3 Limitations Motivating This Thesis

The conclusion of the prior thesis points to four limitations.

1. **Data scarcity in recognition.** Only a handful of short clips where individuals are identified are available, which bottlenecks any face-based identification model. The suggested remedy is a semi-automatic annotation pipeline to scale up the dataset.
2. **Untuned Re-ID for tracking.** OSNet is used out of the box on a pedestrian-Re-ID checkpoint, with no fine-tuning on chimpanzees. Fine-tuning is proposed as the obvious next step.
3. **Tracker choice.** DeepSORT is the only tracker that has been tested. StrongSORT, BoT-SORT and ByteTrack were proposed as candidates for a more rigorous comparison.
4. **Ad-hoc evaluation.** No standard MOT metric is used to quantify tracking quality. Performance is judged qualitatively through the correction interface and indirectly through detection metrics.

These four points map onto the contributions listed in chapter 1 and structure the experimental chapters of this thesis.

Chapter 3

Datasets and Evaluation Infrastructure

Chapter Overview Before any tracker can be trained or evaluated, two kinds of infrastructure must exist and a corpus of accurately labelled video data must be constructed to serve as ground truth. An evaluation pipeline must also be available to produce reproducible quantitative scores. Neither was available off the shelf when we resumed the previous year’s work. This chapter describes how both were built. On the data side, the prior thesis [1] had established a two-step annotation workflow for producing manually corrected tracking files. We extended this workflow to three rendering steps detailed below. Applying this workflow, in collaboration with Calogero Montedoro and Brigitte Rezsohazy, yielded a corpus of eleven fully annotated videos spanning all twenty individuals of the study group. On the evaluation side, HOTA [2] requires per-frame detection sets with integer identity labels and pre-computed pairwise IoU matrices as input. Our annotation files use chimpanzee names as identifiers and a custom `#`-delimited format designed for human readability and direct video rendering. This design was not directly compatible with metric computation. Bridging the two required a dedicated parsing, alignment, and identity-remapping layer.

This chapter introduces the data and tooling that make the experiments of the following chapters possible. It first describes the manual annotation workflow and the resulting ground-truth corpus. It then details the evaluation interface used to compute HOTA and its decompositions under a single reproducible protocol. Finally, it describes the auxiliary datasets mined from the same videos to fine-tune the appearance models used in our ablations.

The same annotated videos also serve a second purpose as they provide a large pool of weakly supervised crops that would otherwise require prohibitive manual labelling effort. Face crops are mined, quality-filtered, and organised into a three-tier training dataset for fine-tuning the ChimpUFE face embedder on our population. In parallel, we assemble an analogous body-crop dataset from the same source videos to fine-tune the OSNet body Re-ID backbone that will later be used in the BoT-SORT ablation discussed in section 4.2.5.

3.1 Manual Annotation Workflow

The prior two-step pipeline produced a raw annotated video from automatic body detection and tracking and a manually corrected version mapping tracker IDs to chimpanzee names [1]. The

three-step extension introduced below builds directly on this design.

3.1.1 Three-Step Rendering Pipeline

The annotation pipeline operates on raw video files and produces progressively refined outputs:

1. **Step 1 - Automated tracking.** Each input video is processed by the YOLOv8s body detector followed by the ByteTrack tracker. The system saves raw tracking output to a text file (one #-delimited block per frame, each line containing ID x1 y1 x2 y2) and renders an annotated video with bounding boxes and frame numbers overlaid. An initially empty manual annotation file is created alongside the output, to be populated by the annotator.
2. **Step 2 - Manual correction.** The annotator watches the Step 1 video and fills in the annotation file, mapping tracker IDs to chimpanzee names. Running Step 2 parses the annotation file, applies the corrections, and relabels each detection accordingly. A new video is rendered with corrected name labels and a frame counter overlay. The frame counter is critical as it allows the annotator to reference specific frames when filling the annotation file.
3. **Step 3 - Final rendering.** Once annotations are verified, Step 3 produces the definitive output. Bounding boxes are replaced by coloured triangle pointers above each individual's head and name labels are displayed prominently. We also decided to remove the frame counter for clarity. This clean rendering serves both as the deliverable for collaborators and as visual confirmation that the ground truth is complete.

All three steps preserve the original audio track of the source video on the rendered output. We made this choice deliberately as the camera operators sometimes call out the chimpanzees' names while filming. These vocal cues and the animals' own vocalisations and environmental sounds help the annotator disambiguate identities in difficult frames.

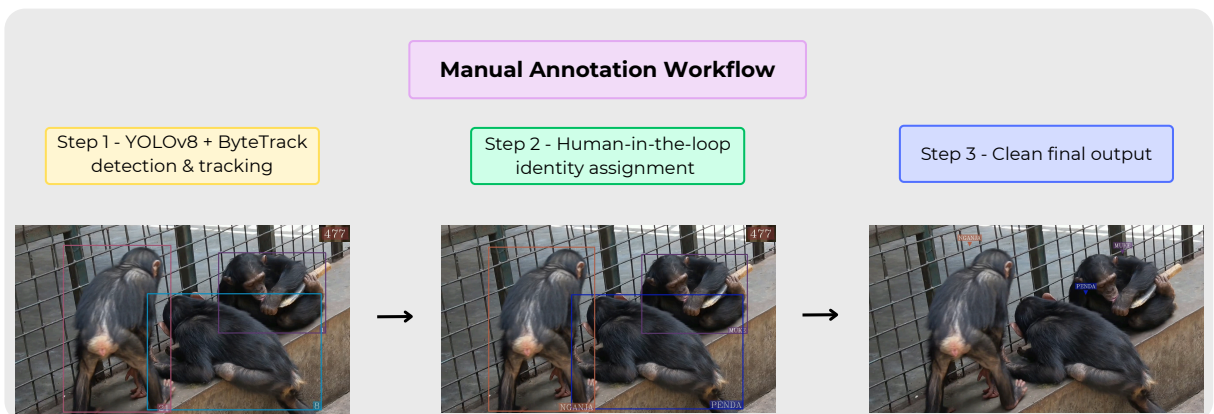


Figure 3.1: Three-step manual annotation workflow applied to chimpanzee footage.



Figure 3.2: Frame 10740 of the 20241018-07h56 video at two stages of the annotation pipeline. **Left:** Step 1 output. Bounding boxes carry arbitrary numeric ByteTrack IDs with no identity information. **Right:** Step 2 output. Ground-truth annotation after manual correction. *Click the image to watch the full comparison clip (1 min 30 s).*

3.2 Ground-Truth Video Corpus

Through the sustained annotation effort of Calogero Montedoro and Brigitte Rezsöházy, five fully annotated ground-truth videos were available by mid-February 2026, and the corpus was later extended to eleven videos by late April 2026. Table 3.1 summarises the corpus.

Curated nature of the annotations. Although the annotation workflow starts from automatic ByteTrack detections, the final ground-truth files should not be interpreted as an exhaustive record of every visible chimpanzee in every frame. During manual correction, the annotator also removed detections that were considered too ambiguous, barely visible, or unnecessary for behavioural interpretation. The resulting ground truth is therefore a curated annotation designed to be readable and useful for downstream biological analysis, rather than a raw preservation of all detector outputs. As a result, the reported tracking metrics should be interpreted as a consistent comparative evaluation across methods under a fixed curation policy, rather than as an absolute measure of detecting every visible chimpanzee in every frame.

3.2.1 Filming Setup and Camera Motion

All ground-truth videos were filmed by a human operator with a handheld camcorder rather than from a fixed mount. The footage therefore contains substantial *global* motion such as hand-held jitter, slow pans following the animals around the enclosure, and frequent optical zoom in and out as the operator reframes a focal individual. From a tracking perspective, the apparent inter-frame displacement of a chimpanzee is therefore the superposition of its own motion in world

Table 3.1: Ground-truth video corpus. Each video is fully annotated with per-frame identity bounding boxes. *Outside* videos were filmed from outside the enclosure and the wire-mesh barrier is omnipresent in the images. *Inside* videos were filmed without obstruction from within the enclosure. The Individuals column reports the number of distinct chimpanzee identities appearing at least once in the annotated file.

Video ID	Video Name	Annotation Date	Camera View	Individuals	Frames	Role
A	07h56	09/02/2026	Inside	14	55 847	Train
B	12h28	27/02/2026	Outside	6	53 542	Train
C	06h17	27/02/2026	Outside	4	34 967	Train
D	13h28	27/02/2026	Inside	3	55 127	Train
E	14h29	27/02/2026	Inside	3	55 030	Train
F	08h05	09/04/2026	Inside	6	45 214	Train
G	12h19	09/04/2026	Inside	14	46 119	Train
H	13h19	09/04/2026	Inside	11	47 134	Train
I	06h57	09/04/2026	Inside	12	55 622	Train
J	12h33	16/04/2026	Inside	20	47 422	Eval
K	11h33	23/04/2026	Inside	19	46 342	Eval

coordinates and the camera’s motion in image coordinates. A constant-velocity Kalman filter operating purely in pixel space cannot separate the two, and even moderate camera pans or zooms can push a predicted box several tens of pixels away from the next true detection of the same individual. We will return to this point when interpreting the tracker comparison (Sections 5.3.2 and 5.3.3), since it directly affects how camera-motion compensation (CMC/GMC) and strict IoU gating should be read.

3.2.2 Selection of Evaluation Videos

While the videos from A to I were used for training body and face appearance models (to maximise the available data), the two last videos (J and K) were reserved for the final pipeline evaluation. Because videos J and K were annotated after the training corpus had been fixed, they provide an evaluation set with no overlap with the data used to train the appearance models. Both videos contain dense multi-individual indoor scenes with frequent occlusions and social interactions. Video J contains the full group of 20 individuals, whereas video K contains 19 individuals, with Kalemi absent.

J: 12h33		K: 11h33	
<i>Number of Individuals</i>	20	<i>Individuals</i>	19
<i>Number of Frames</i>	47 422	<i>Frames</i>	46 342
<i>Duration</i>	15 min 48 s	<i>Duration</i>	15 min 26 s
<i>Frame rate</i>	50 fps	<i>Frame rate</i>	50 fps

Outside videos were retained for training diversity, but the final evaluation was restricted to

inside videos because the omnipresent wire mesh in outside footage introduces a distinct visual domain.

3.3 Preparing Data for HOTA Evaluation

The annotated videos are only indirectly useful for quantitative evaluation. HOTA operates on per-frame sets of detections with associated identities and on the pairwise spatial similarities between them [2], not on annotated video files. This section describes the thin data layer that bridges the annotation workflow and the HOTA engine. The mathematical definition of HOTA was given in section 2.4.4.

3.3.1 A Single Interchange Format for Ground Truth and Predictions

Both the manually corrected ground truth and every tracker output are serialised in the same lightweight text format. A file represents one video as a sequence of frame blocks separated by a line containing only the character `#`. Within a frame block, each non-empty line encodes one bounding box:

ID x1 y1 x2 y2

where `ID` is the trajectory label and (x_1, y_1, x_2, y_2) are the absolute pixel coordinates of the top-left and bottom-right corners in the source video. An excerpt from a ground-truth file for video **A** is shown below. The corresponding prediction file has identical structure but with tracker-assigned integers in place of names.

Listing 3.1: Two consecutive frames in the custom HOTA interchange format. Ground-truth identities are chimpanzee names. Prediction identities are tracker-assigned integers. The geometry and frame structure are identical in both cases.

#				
Amadi	290.88	8.21	617.04	398.30
Binasera	826.90	352.53	1310.95	735.27
#				
Amadi	338.95	4.43	627.25	349.28
Binasera	839.08	356.79	1316.38	734.31

Three design choices are worth highlighting. First, the representation is deliberately symmetric. The only difference between a ground-truth file and a prediction file is semantic (names vs. tracker IDs), not structural. The evaluator therefore remains agnostic to the tracker that produced the predictions, which is essential for the cross-tracker comparison reported in Chapter 5. Second, coordinates are stored as absolute corner pixels rather than the normalised centre-width-height tuples used by YOLO training files. This matches the output space of the trackers (whose bounding boxes are already in pixel coordinates) and makes any file directly renderable on top of the source video for debugging. Third, empty frames are represented explicitly by consecutive delimiters. Preserving them is mandatory as skipping empty frames would desynchronise the ground-truth and prediction sequences and silently invalidate every subsequent frame-wise comparison.

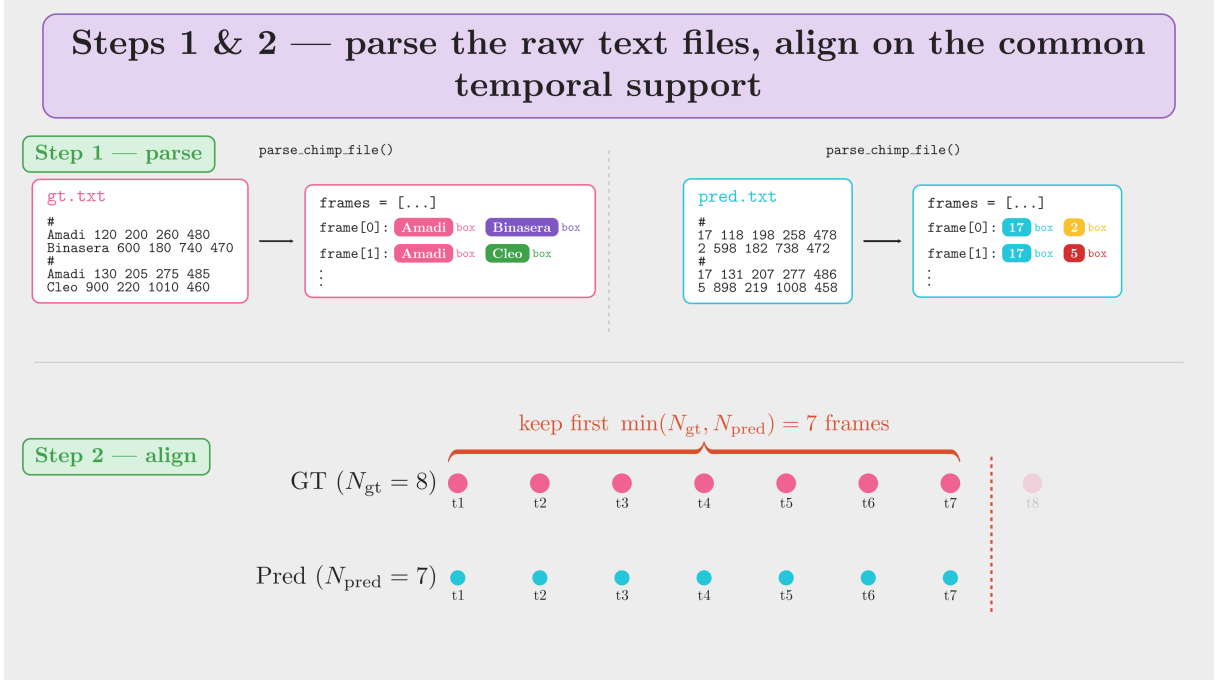


Figure 3.3: Steps 1 and 2 of the preparation pipeline. **Step 1 (parse)**. A single parser reads both the ground-truth file (left) and the prediction file (right) into the *same* ordered list-of-frames structure. Only the identity labels differ (chimpanzee names on the GT side, tracker integers on the prediction side). **Step 2 (align)**. The two sequences are then truncated to their common temporal support.

3.3.2 From Text Files to HOTA Inputs

Given a pair of files, the evaluator produces the data structure expected by the HOTA formulation in four steps.

Step 1 - Parsing. Each file is parsed into an ordered list of frames. A frame is itself a list of detections and each detection retains only its label and its bounding box. Blank lines are discarded, but frame delimiters are preserved so that empty frames are materialised as empty lists. This preserves the one-to-one correspondence between list index and frame index.

Step 2 - Temporal alignment. Evaluation is restricted to the common temporal support of the two sequences. If the ground-truth file contains N_{gt} frames and the prediction file contains N_{pred} frames, only the first $\min(N_{gt}, N_{pred})$ frames are retained on both sides. In practice both files originate from the same source video and are generally of equal length, but this conservative truncation protects against off-by-one mismatches at the very end of a sequence (for instance when a tracker drops the last frame).

Step 3 - Independent identity remapping. Textual identity labels are not consumed directly by the HOTA engine as it expects contiguous integer indices for efficient matrix indexing. The set of ground-truth labels is therefore collected from all frames and mapped to $\{0, 1, \dots, N_{gt-id} - 1\}$. The set of predicted labels is mapped independently to $\{0, 1, \dots, N_{pred-id} - 1\}$. The two mappings are *disjoint by construction* and not aligned in any way.

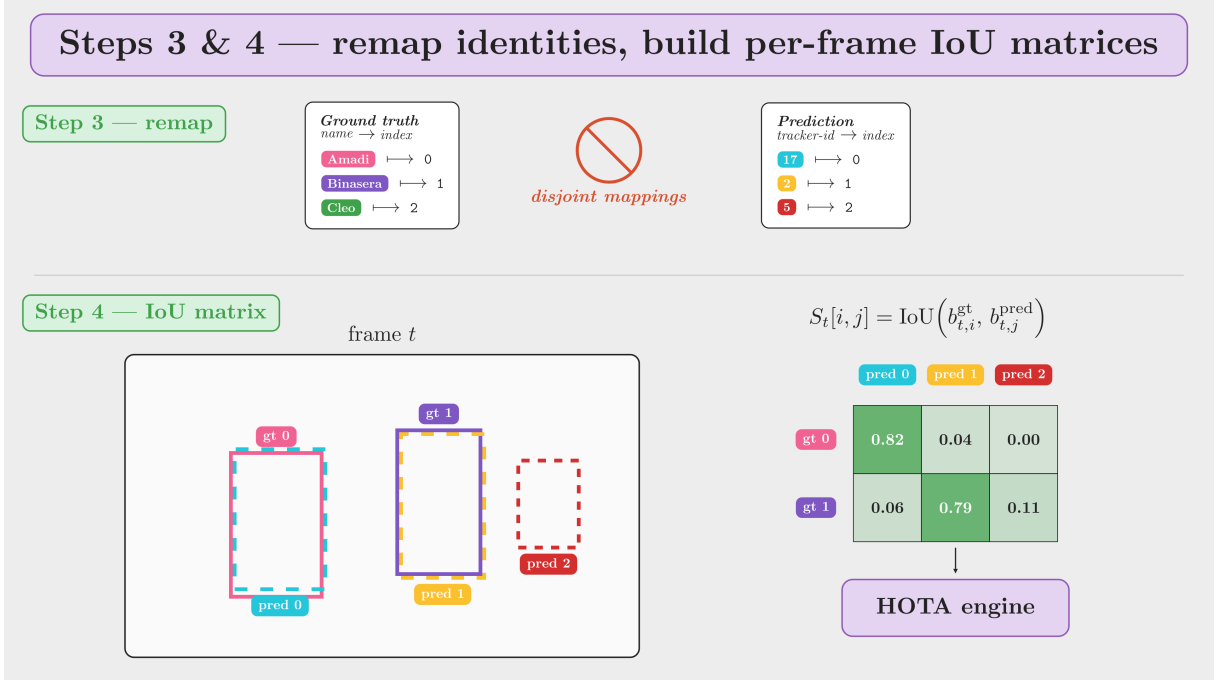


Figure 3.4: Steps 3 and 4 of the preparation pipeline. **Step 3 (remap)**. The set of ground-truth labels and the set of predicted labels are compressed independently into contiguous integer indices. **Step 4 (IoU matrix)**. For every retained frame t , the evaluator builds a $m_t \times n_t$ intersection-over-union (IoU) matrix between every ground-truth box (*solid*) and every predicted box (*dashed*). Darker cells correspond to higher IoU. The two near-diagonal entries (0.82, 0.79) are the matches that Hungarian assignment will recover, while the third predicted box (**pred 2**) produces a column of small values that HOTA will flag as a false positive.

This independence is conceptually important and is often a source of confusion. HOTA does not require (and does not attempt to leverage) any lexical correspondence between ground-truth names (e.g. **Amadi**) and tracker-assigned integers (e.g. 17). Instead, it recovers the correspondence through per-frame Hungarian matching and the global association score, as detailed previously in section 2.4.4. The remapping step merely compresses each identity space into contiguous indices while preserving its internal consistency and this is exactly what the metric needs.

Step 4 — Per-frame similarity matrices. For each retained frame t , containing m_t ground-truth and n_t predicted detections, the evaluator computes an $m_t \times n_t$ intersection-over-union (IoU) matrix between all ground-truth and predicted boxes:

$$S_t[i, j] = \text{IoU}(b_{t,i}^{\text{gt}}, b_{t,j}^{\text{pred}}), \quad i \in \{1, \dots, m_t\}, j \in \{1, \dots, n_t\}. \quad (3.1)$$

The per-frame IoU matrices and the two lists of remapped identity arrays constitute the complete input to the HOTA engine. They are exactly the objects over which the localisation thresholds $\alpha \in \{0.05, 0.10, \dots, 0.95\}$ are swept and over which the global alignment score and the TPA/F-NA/FPA counts of section 2.4.4 are accumulated by averaging (see eq. (2.14)).

The final scalar output is not restricted to HOTA alone. The run is summarised by the seven metrics used throughout this thesis (HOTA, DetA, AssA, DetRe, DetPr, AssRe, and

AssPr) and appended to `history.json` with the video name, the `model_version` tag, and a timestamp. This structured log is what enables the cross-tracker, hyperparameter, and post-processing comparisons reported in Chapter 5: every result table ultimately traces back to an entry in this single file.

3.3.3 Identity-Aware Recognition Metrics

HOTA measures association quality from the point of view of arbitrary track identifiers and is invariant to which chimpanzee name is attached to a given track. To complement it, the final auto-labelling pipeline also emits two identity-aware metrics that operate directly on the predicted names, plus an abstention rate that quantifies the third possible outcome of a matched prediction.

Let $\mathcal{G}$ denote the set of ground-truth face detections with their identity labels, and $\mathcal{P}$ the set of predicted boxes produced by the pipeline (each carrying either a chimpanzee name $c \in \{1, \dots, N_C\}$ or the special label `unknown_n` for tracklets that the recognition stage refused to commit to). Let

$$\mathcal{M} = \{(g, p) \in \mathcal{G} \times \mathcal{P} : \text{IoU}(g, p) \geq \tau_{\text{IoU}}\} \quad (3.2)$$

be the set of one-to-one spatial matches between ground-truth and predicted boxes, computed with the same IoU threshold $\tau_{\text{IoU}} = 0.5$ as the HOTA evaluation. For a matched pair $(g, p) \in \mathcal{M}$, denote by $y(g)$ the ground-truth identity and by $\hat{y}(p)$ the predicted label.

Identification Accuracy (IDAcc). IDAcc is the fraction of matched predictions whose label is the correct chimpanzee name:

$$\text{IDAcc} = \frac{|\{(g, p) \in \mathcal{M} : \hat{y}(p) = y(g)\}|}{|\mathcal{M}|}. \quad (3.3)$$

It measures the *purity* of the names that the pipeline emits and ignores ground-truth boxes that received no spatial match.

Identification Recall (IDRec). IDRec is the fraction of ground-truth detections that receive both a spatial match and the correct label:

$$\text{IDRec} = \frac{|\{(g, p) \in \mathcal{M} : \hat{y}(p) = y(g)\}|}{|\mathcal{G}|}. \quad (3.4)$$

Compared with IDAcc, IDRec penalises both spatially unmatched ground-truth boxes and matched-but-misnamed predictions, and therefore captures the *coverage* of the named annotation.

Unknown rate. Among the matched predictions, a third category receives the abstention label `unknown_n`, typically because the tracklet failed the minimum-length gate or because the cluster-level recognition stage left it isolated from every known identity. The unknown rate is

$$\text{Unknown} = \frac{|\{(g, p) \in \mathcal{M} : \hat{y}(p) = \text{unknown_n}\}|}{|\mathcal{M}|}. \quad (3.5)$$

By construction, the three quantities satisfy $\text{IDAcc} + \text{Unknown} + \text{Misnamed} = 1$ on the matched subset, where $\text{Misnamed} = 1 - \text{IDAcc} - \text{Unknown}$ is the fraction of matched predictions that are spatially correct but assigned to the wrong chimpanzee. This decomposition makes the error budget more informative than IDAcc alone, since it separates active misnaming from conservative abstention. The numerical values reported for the evaluation pair are collected in Table 5.8.

3.4 Body Re-Identification Dataset

To support the BoT-SORT ablation of section 4.2.5, in which we test whether a chimpanzee-specific appearance backbone improves tracker association, we assembled a body-crop dataset for fine-tuning the OSNet Re-ID model [17]. Body crops are mined directly from the ground-truth annotated videos, reusing the named bounding boxes as identity labels at no additional annotation cost.

The dataset was built in February 2026, when only videos **A–E** were annotated. These five videos cover 18 of the 20 identities of the study group. Crops are sampled every fifth frame, filtered for minimum size and sharpness, and capped at 2000 crops per identity per source video to limit class imbalance. A 70/30 temporal split within each video yields **37 134 crops** (29 724 training, 7 410 validation) with the per-identity distribution shown in Figure 3.5. The corpus was not regenerated after the ground-truth corpus was extended, because the BoT-SORT ablation had by then already established that body Re-ID is not the bottleneck (see sections 5.3.2 and 5.3.3).

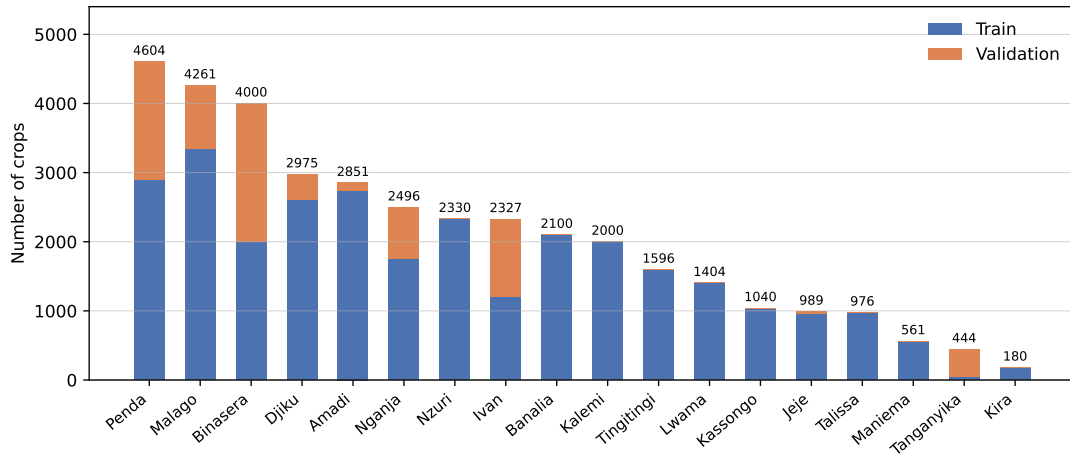


Figure 3.5: Class distribution of the body Re-ID dataset across the 18 identities visible in videos A–E.

3.5 Face Embedder Fine-Tuning

3.5.1 Motivation

One of the main limitations of the face-recognition models developed in the previous thesis was the small number of training images available for each individual. In this section, we explain

how we addressed that limitation by expanding the training set by a factor of twenty through automatic mining of additional face crops from annotated tracking videos. We then describe how the mined data was structured into quality tiers, so that the cleanest samples provide a strong identity signal while still allowing noisier samples to contribute useful variability. This tiered dataset was subsequently used to fine-tune the ChimpUFE face embedder for our population of twenty chimpanzees (see section 4.3).

3.5.2 Three-Tier Dataset Overview

The training set draws on three sources of decreasing quality and increasing size. The first is composed of high-quality images extracted from last year’s **ChimpRec** database. The second is a few hundred crops manually annotated by students during trombinoscopy practical sessions. The third comes from the annotated tracking videos of the ground-truth corpus. For this last tier, we run the pretrained YOLOX face detector of the *ChimpUFE* project, on those footage and inherit the identity of the surrounding body track. This yields about twenty times more samples than the other two sources combined, but also inherits errors from both the tracker and the face detector.

To keep the noisy mined crops from drowning out the cleaner sources, we treat the three sources as separate *tiers* rather than pooling them into a single training set. This tiered organisation makes it possible, on the method side, to control independently how often each source contributes to a training batch and how much weight its samples carry in the loss. The concrete mechanisms used to balance the tiers, together with the experiments that motivated them, are method-side concerns and are described in section 4.3.

Table 3.2: Three-tier dataset structure used for ChimpUFE fine-tuning. N_t is the total number of images in tier t , including both training and validation images. *Quality* is a qualitative summary of how clean the labels and the crops are.

Tier	Source	N_t	Quality
1	ChimpRec curated crops	1093	High
2	Student trombinoscopy	522	Medium/High
3	Mined from videos	18 225	Low
Total	—	19 840	—

3.5.3 Tier 1 ChimpRec Database

Tier 1 is the highest-quality source and originates from last year’s **ChimpRec** database [1]. Images were hand-picked and labelled with canonical identity names. Head localisation labels were used to extract tighter face crops. The split is fixed and reused across all fine-tuning experiments to provide a stable validation signal. Tier 1 contains **866 training** and **227 validation images** evenly distributed across the 20 individuals. Per-individual counts are reported in Appendix D.

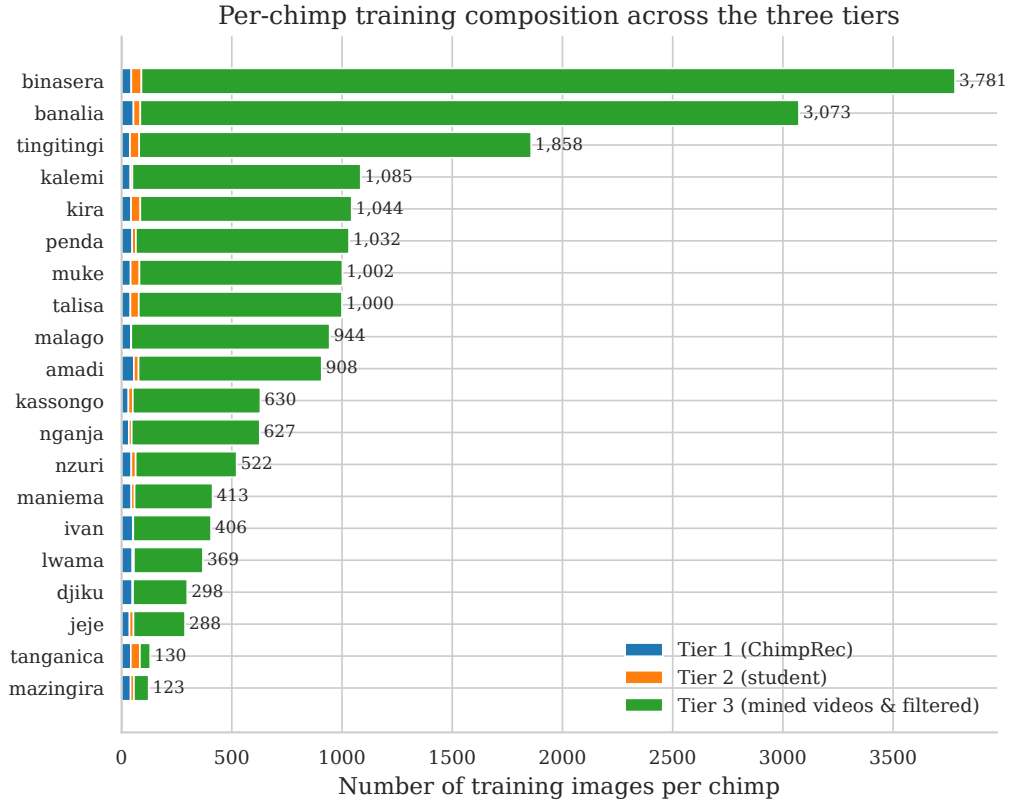


Figure 3.6: Per-chimp image counts, including training and validation images, in each tier.

3.5.4 Tier 2 Student Trombinoscope

Tier 2 was collected during trombinoscopy practical sessions in February 2026. Students annotated identity on face crops extracted from pre-boxed videos, and uncertain cases were discarded. Crop and label quality is more variable than Tier 1, but Tier 2 increases pose and illumination diversity. The corpus contains **522 images** across 20 individuals with an imbalanced class distribution. We apply a deterministic per-individual 85/15 split (at least one validation image per class). Per-individual counts are reported in Appendix D.

3.5.5 Tier 3 - Mined Face Crops from Annotated Videos

Tier 3 uses only videos A–I to mine additional face crops with no annotation cost, while videos J and K are excluded from all training data and reserved for evaluation. The idea is to use the fact that the named tracking files already associate every body bounding box with a chimpanzee identity, so we can run a face detector on the same footage and inherit the identity of the surrounding track.

Automatic Face Crop Mining

Each frame is processed with the pretrained YOLOX face detector provided in the *ChimpUFE* project repository [26]. We initially considered reusing the face detector from the previous ChimpRec pipeline [1]. Qualitative inspection on our footage showed that it produced less reliable face crops for mining, so we adopted the *ChimpUFE* detector.

This automatic mining procedure produced **30 502 raw crops**. However, many of these crops are not directly useful for face recognition. Some show the back of the head, some are too blurry or too dark, and some are false detections such as hands or background regions. Training directly on all these crops would add too much noise to the face embedder. We therefore added a heuristic filtering step before using the mined crops for training.

Quality and Frontal Face Filtering

The goal of the filtering step is not to perfectly estimate the pose of the chimpanzee, but only to remove the most obvious bad crops and to keep crops that are more likely to contain a visible face looking approximately towards the camera. The filter is based on three elements: basic image quality, approximate frontal orientation, and weak eye visibility.

Image quality filtering. Each crop is first checked with simple image statistics. We remove crops that are too small, too blurry, too dark, too bright, or have too little contrast. Blurriness is estimated using the variance of the Laplacian on the grayscale image, a common focus-measure operator [27], since sharp images contain stronger high-frequency variations. In the final pipeline we require a minimum width and height of 90 pixels, a minimum Laplacian variance of 120, a minimum grayscale contrast of 28, and a mean brightness between 35 and 220. All thresholds were chosen empirically through iterative visual inspection on a development subset of mined crops drawn from training videos only. We adjusted the values to remove clearly unusable detections while preserving pose and illumination diversity among the retained faces. Once fixed, the thresholds were kept unchanged for all subsequent mining runs.

Approximate frontal score. For the crops that pass the quality filter, we compute an approximate frontal score $f \in [0, 1]$. This handcrafted score combines several simple cues that we found useful on chimpanzee face crops. A frontal face tends to be more symmetric between the left and right halves, which is captured by the left-right symmetry term sym_{LR} . It also tends to have more balanced brightness between the two sides, measured by the brightness-balance term $\text{bright}_{\text{bal}}$, and more balanced edge structure on both sides, measured by the gradient-balance term grad_{bal} . Frontal faces also tend to be better centred in the crop, which is captured by the centring term center . We additionally include a chimpanzee-specific cue. When the face is frontal, the central muzzle and brow region is often brighter than the surrounding dark fur. This effect is captured by $\text{skin}_{\text{patch}}$, which compares the brightness of a central elliptical region with that of a surrounding ring. The final score is computed as a weighted sum of all those features. In the final Tier 3 pipeline, we keep only crops with $f > 0.55$.

$$f = 0.40 \cdot \text{sym}_{\text{LR}} + 0.25 \cdot \text{skin}_{\text{patch}} + 0.15 \cdot \text{grad}_{\text{bal}} + 0.12 \cdot \text{center} + 0.08 \cdot \text{bright}_{\text{bal}}. \quad (3.6)$$

Final selection. Only crops that pass the quality filter, the frontal-score threshold, and the eye-visibility constraint are kept. This filtering step reduces the raw mined dataset from **30 502 crops** to **18 225 crops**. The retained crops form the final Tier 3 dataset used during training.

Because consecutive mined crops are noisier than the manual sources, Tier 3 was used only for training and not for validation. The validation signal was therefore kept cleaner by relying on the manually annotated tiers, while Tier 3 was used to increase the amount and diversity of training samples.

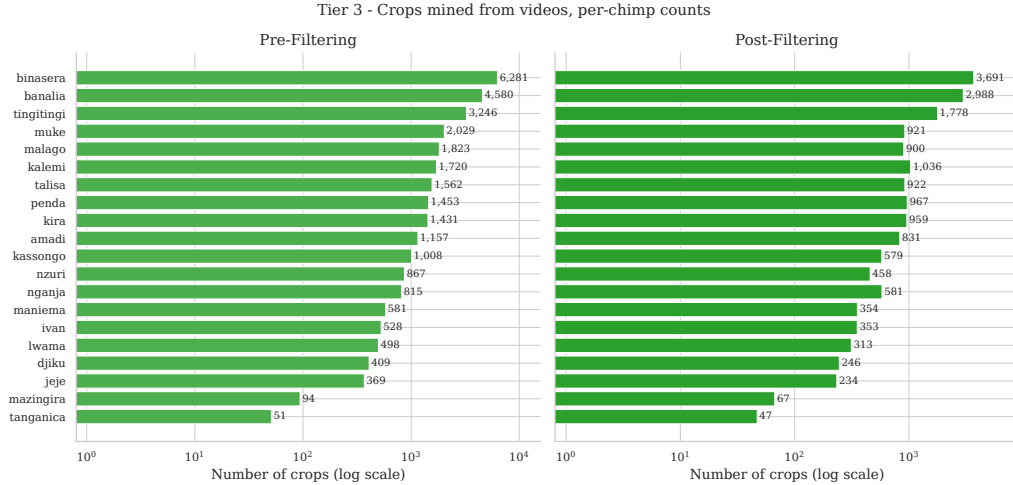


Figure 3.7: Per-individual image counts in the Tier 3 mined dataset, before and after heuristic frontal face filtering.

3.5.6 Test Set

To evaluate the fine-tuned face embedder, we assembled a dedicated test set composed of face crops with controlled framing but variable visual quality. The initial version of this set was built from the **ChimpRec** database of the previous year, yielding a total of **132 images**. Although these images are generally well cropped and of good visual quality, the resulting dataset was imbalanced, with several classes underrepresented and some individuals not present at all.

To obtain a more reliable and representative evaluation, we extended this initial set with additional manually curated face crops. These supplementary images were extracted from the annotated ground-truth videos J: 12h33 and K: 11h33 (Table 3.1). These two videos were selected for two reasons. First, they were not used at any stage of the face embedder training pipeline. Second, they contain all individuals of the study group.

Although all chimpanzees are present in these videos, some individuals (Kalemi, Mazingira, Nzuri, Ivan, Talisa, and Tingitingi) appear only in a limited number of frames and do not exhibit sufficient pose diversity to be considered meaningful. With the help of Brigitte Rezsöházy, we extracted additional face crops for these individuals from other videos that had not yet been processed (20241009 - 08h04, 20241218 - 11h19, 20241012 - 09h00, 20241006 - 07h04).

In the end, we obtained 25 images per chimpanzee, resulting in a total of **500 images** in the final test set. The additional crops were manually selected from different time segments and under a wide range of poses and viewpoints. This ensures that the test set reflects realistic video conditions, including variations in head pose, occlusion, and illumination. The final test set therefore combines a uniform distribution across all identities with a high intra-class diversity

while preserving a broad range of image quality.



Figure 3.8: Examples of face crops of Ivan illustrating the variability of visual quality in the test set. From left to right: a high-quality frontal face, a medium-quality example affected by moderate blur, and a low-quality crop with strong pixelation or degradation.

This test set is used exclusively for the final evaluation of the fine-tuned model and is not involved in any part of the training or validation process.

Chapter 4

Methods

Chapter Overview. This chapter describes the experimental protocols used throughout the thesis. It first specifies the hardware and software environment used for all experiments. It then presents the common tracking protocol and the tracker variants compared in the study. Next, it details the procedure used to fine-tune ChimpUFE for chimpanzee face identification. Finally, it introduces the two post-processing pipelines that convert tracker-generated body tracklets into identity-labelled trajectories.

4.1 Experimental Hardware and Runtime Environment

All experiments reported in this thesis were conducted on the *Lyra* cluster of the CECI (Consortium des Équipements de Calcul Intensif) using the SLURM workload manager. Jobs were submitted to GPU-enabled nodes and executed under identical hardware conditions.

Compute node configuration. Each job was allocated a single compute node with the following characteristics:

- **CPU:** AMD EPYC (Genoa) processor with 16 physical cores.
- **RAM:** 62 GB of system memory.
- **GPU:** NVIDIA RTX 6000 Ada Generation with 48 GB VRAM.

Software environment. All experiments were run under the following software stack:

- **Operating system:** Rocky Linux 9.5.
- **Kernel:** Linux 5.14.
- **Python:** 3.10.13.
- **PyTorch:** 2.7.1 with CUDA 12.8.

4.2 Tracking Methods

4.2.1 Common Tracking Protocol

All four trackers compared in this thesis are tracking-by-detection systems built on top of the same YOLOv8s [1] body detector. To make their comparison fair, every tracker is exercised

under the same cached-replay protocol. YOLOv8s detections are first computed once per video at a low confidence floor, saved to disk, and then replayed frame by frame through each tracker. For the appearance-based trackers (StrongSORT and BoT-SORT), the corresponding Re-ID embeddings are pre-computed in the same way. Detection inputs are therefore identical across configurations, and the cached and uncached paths produce numerically identical predictions for any fixed tracker setting.

Tracker hyperparameters are tuned on videos *A* (07h56, 14 individuals) and *E* (14h29, 3 individuals), the two earliest fully annotated indoor recordings. They jointly cover a dense and a sparse-density regime. Every configuration is scored by the HOTA pipeline described in the previous chapter in section 3.3, and the optimisation criterion is the mean HOTA across the two tuning videos. The evaluation videos J and K are never used during tuning and are reserved for the evaluation-video comparison.

The three tuned trackers (StrongSORT, BoT-SORT, and ByteTrack) share the same two-phase search strategy. A uniform random exploration of the grid is performed followed by a local refinement around the best Phase 1 candidates. This chapter defines the common protocol and the role of each tracker. The exhaustive search spaces, selected values, and diagnostic plots are collected in Appendix B. Their interpretation is presented in the results chapter.

4.2.2 Inherited DeepSORT Baseline

The starting point of our work is the ChimpRec pipeline, which combines YOLOv8s body detection with DeepSORT and a pedestrian-pretrained `osnet_x1_0` appearance backbone. Since the face-recognition branch of the previous thesis reached only 13.6% top-1 accuracy, we disabled both face detection and face recognition in the baseline, leaving DeepSORT alone on top of the body detector. The baseline was not retuned and is used at its shipped configuration.

4.2.3 StrongSORT Integration

StrongSORT is included as a modernised DeepSORT variant. Relative to DeepSORT, it replaces the gallery of recent appearance descriptors with an EMA appearance state, introduces a confidence-aware Kalman filter, and integrates an optional camera-motion compensation step [11]. We use the same human-pretrained `osnet_x1_0` backbone as the DeepSORT baseline so that the comparison isolates StrongSORT’s motion- and gallery-side changes rather than a new appearance model. Camera-motion compensation was disabled to keep the online-tracker comparison focused on the same detector and image-coordinate motion model. Seven hyperparameters (detection confidence, EMA decay, cascade lifetimes, cosine and IoU gates) were tuned on the cached detections and embeddings.

4.2.4 ByteTrack as a Motion-Only Tracker

ByteTrack [14] is the motion-only branch of the comparison and the upstream tracker used by the production pipeline. It deliberately drops the appearance branch and relies on Kalman-based motion prediction and IoU association. Its distinguishing feature is the two-stage BYTE

matching procedure. High-confidence detections are first matched against active tracks, then *low-confidence* detections are matched against the still-unassigned tracks in a second pass. Detections that would normally be discarded as noise are therefore recovered when they fall close to a predicted track box and that is exactly the regime in which a partially occluded chimpanzee briefly drops below the detector confidence threshold.

Seven parameters were tuned with the shared two-phase protocol, the two association thresholds, the new-track threshold, the track buffer, the IoU match gate, the score-fusion flag, and the detection confidence. The selected configuration is reported in Appendix B and is kept fixed for all downstream post-processing experiments.

4.2.5 BoT-SORT Re-ID Ablation

BoT-SORT extends ByteTrack’s two-stage IoU association with an appearance gate while keeping the rest of the association logic close to ByteTrack. This makes it the cleanest tracker for testing whether a chimpanzee-specific body Re-ID model improves online association because enabling and disabling the appearance branch changes one component at a time. We selected it over StrongSORT for this ablation because it slots into the existing Ultralytics cached-replay pipeline and allows the same ByteTrack motion parameters to be reused.

The five motion-related parameters were frozen at the ByteTrack parameter optimum, and BoT-SORT’s global motion compensation was disabled so that the ablation remained in the same image-coordinate motion regime as ByteTrack. The five appearance-related or shared parameters (proximity gate, appearance gate, IoU match, track buffer, and detection confidence) were then tuned with the appearance branch both enabled and disabled. To give Re-ID its best chance, we fine-tuned OSNet on the chimpanzee body corpus defined in the previous chapter section 3.4 and used the best validation Rank-1 checkpoint.

4.3 Fine-Tuning ChimpUFE with an ArcFace Identity Head

4.3.1 Overview

In this section, we describe how the three-tier corpus is used to fine-tune the ChimpUFE pre-trained backbone into our ChimpUFE fine-tuned model. The main difficulty is that Tier 3 contains many more images than Tier 1 and Tier 2, but its labels are also less reliable. If all images were sampled uniformly, training would therefore be biased toward automatically mined video crops.

To avoid this, we explicitly control both the sampling frequency and the loss contribution of each tier. Cleaner images are made more influential, and Tier 3 remains useful for increasing visual diversity without dominating the training. This is done with a tier-aware sampler, per-sample loss weighting, and tier-specific augmentation policies.

The fine-tuning itself is performed in two stages. We first train only the ArcFace head while keeping the ChimpUFE pretrained backbone frozen, and then fine-tune the full network (pretrained backbone + ArcFace head) with a smaller learning rate for the backbone.

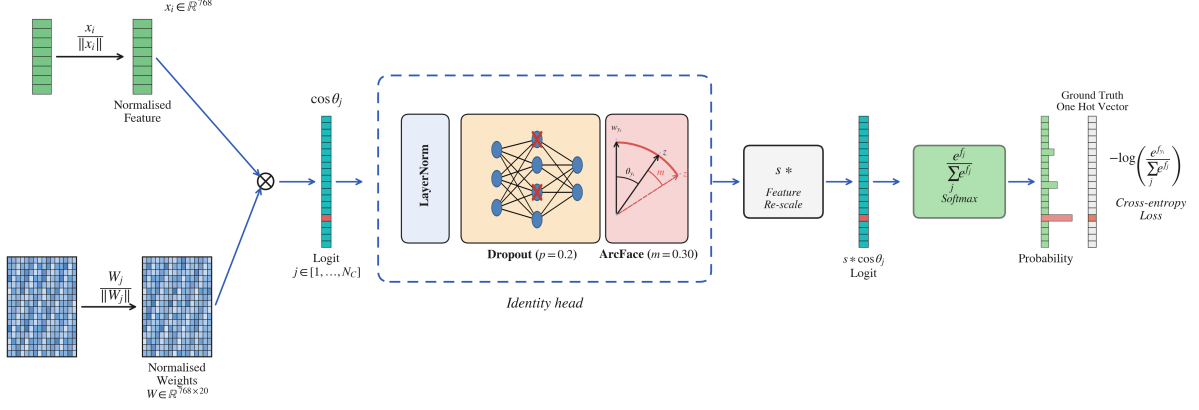


Figure 4.1: Overview of the fine-tuning pipeline. The ChimpUFE pretrained backbone extracts one embedding per face crop. This embedding is then classified by the ArcFace head, which uses learnable class prototypes and an angular margin. The resulting logits are passed to the cross-entropy loss. During the first stage, only the head is updated and during the second stage, gradients are also propagated through the backbone.

4.3.2 Backbone and Identity Head

Backbone. We start from the publicly available ChimpUFE checkpoint [3], which is a ViT-B/14 model that was pretrained in a self-supervised way (DINOv2 [20]) on a large chimpanzee face dataset. From this checkpoint we keep the ChimpUFE pretrained backbone and drop the projection head that was used during self-supervised pretraining, since it is not useful for our identification task. To avoid introducing an additional interpolation step for the positional embeddings, we stayed as close as possible to the conditions under which the backbone was pretrained. We keep the same input resolution of 224×224 and the same patch size of 14. The network outputs a 768-dimensional embedding $z \in \mathbb{R}^{768}$.

Identity head. On top of the backbone, we add a head that maps the 768-dimensional embedding to the $N_C = 20$ chimpanzee identities of our corpus. The head consists of a **LayerNorm**, a **Dropout** layer with $p = 0.2$, and an ArcFace classifier.

The ArcFace classifier contains one learnable weight vector w_c per identity c , which can be interpreted as a class prototype. During the ArcFace computation, both the embedding z and the prototypes w_c are ℓ_2 -normalised. As a result, the classifier compares an image and a class through their angular similarity on the unit hypersphere. The final class scores are then computed using the ArcFace angular-margin logits introduced in the background chapter in eq. (2.1). Figure 4.1 summarises the fine-tuning process with the ArcFace identity head.

4.3.3 Two-Stage Fine-Tuning Schedule

We fine-tune the model in two stages. This schedule is designed to first adapt the classifier to our 20 chimpanzee identities without disturbing the pretrained representation, and only after that first step, allow the full ViT backbone to move.

First stage. The ChimpUFE pretrained backbone is frozen and only the identity head is trained. This is equivalent to a linear probing stage, except that our classifier is an ArcFace head instead of a linear softmax classifier. The goal of this stage is to initialize the class prototypes on top of the backbone and to let the head move freely without introducing large modifications in the already useful ChimpUFE embedding space.

Second stage. The whole network is unfrozen and fine-tuned end-to-end. The ChimpUFE backbone is trained with a smaller learning rate than the head:

$$\eta_{\text{backbone}} = 2 \times 10^{-5}, \quad \eta_{\text{head}} = 5 \times 10^{-4}. \quad (4.1)$$

It allows the classification head to continue adapting quickly while the DINOv2 backbone is only adjusted slowly. This distinction is important because the backbone already contains a strong self-supervised representation of chimpanzee faces, and we only want to specialize it to our 20 known individuals rather than overwrite it completely.

Both stages use the same tier-aware sampler, the same per-sample loss weighting, and the same validation protocol. The learning rate schedule is reset at the beginning of the second stage so that the full fine-tuning stage has its own warmup and cosine decay.

4.3.4 Tier-Aware Sampling and Loss Weighting

As introduced in the previous chapter, the fine-tuning dataset combines three tiers with very different sizes and label reliability. Tier 3 contains many more images than Tier 1 and Tier 2, but it is also the least reliable source because identities are inherited from automatic video mining. If training sampled images uniformly, optimisation would be dominated by Tier 3 and would amplify its label noise.

To prevent this, we keep the tier label of each image and use it in two complementary ways. First, the tier controls how often samples are drawn through a *sampling multiplier* m_t . Second, the tier controls how strongly each sampled image influences the objective through a *loss weight* λ_t . In addition, we apply a global resampling cap that limits excessive reuse of a small number of images for classes that contain fewer crops.

Tier-aware and class-balanced sampling. Let i be a training image, with identity y_i and tier t_i . We assign it an unnormalised sampling weight

$$w_i = \frac{m_{t_i}}{N_{y_i}}, \quad (4.2)$$

where N_{y_i} is the number of training images of identity y_i across all tiers. The factor $1/N_{y_i}$ compensates for class imbalance, while m_{t_i} increases the visibility of cleaner tiers.

In the final run, we use $m_1 = 5.0$, $m_2 = 3.0$, and $m_3 = 1.0$, so Tier 1 is sampled more often than Tier 2, and Tier 2 more often than Tier 3 after class balancing.

Global resampling cap. Class-balanced sampling can oversample a small number of images for under-represented identities, which increases the risk of memorisation. We therefore cap the expected reuse of any single image within one epoch. In the final run, the cap is set to $R = 3.0$, which prevents any crop from being sampled excessively.

Tier-weighted loss. After sampling and augmentation, the model produces one cross-entropy loss per image. We scale each per-image loss by a tier coefficient λ_{t_i} so that noisier tiers contribute less to the optimisation. For a batch $\mathcal{B}$, the objective is

$$\mathcal{L}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \lambda_{t_i} \text{CE}_\varepsilon(\ell_i, y_i), \quad (4.3)$$

where ℓ_i is the ArcFace logit vector and CE_ε is cross-entropy with label smoothing $\varepsilon = 0.1$.

In the final run, we use $\lambda_1 = 1.0$, $\lambda_2 = 1.5$, and $\lambda_3 = 0.045$. This keeps Tier 3 useful for visual diversity while strongly limiting its influence compared to manually labelled tiers.

Resulting tier influence. Table 4.1 summarises the chosen hyperparameters and the resulting expected tier influence after applying class balancing and the resampling cap. The detailed computation of the expected tier contributions is provided in Appendix D.1.

Table 4.1: Tier-specific sampling multipliers m_t and loss weights λ_t used in the final fine-tuning run. The column $m_t \lambda_t$ gives a simplified per-sample weighting before class balancing and resampling. The column ρ_t reports the expected loss-weighted sampling contribution after applying the full sampling strategy (Appendix D.1).

	m_t	λ_t	$m_t \lambda_t$	ρ_t
Tier 1	5.0	1.000	5.000	0.510
Tier 2	3.0	1.500	4.500	0.339
Tier 3	1.0	0.045	0.045	0.151

With these parameters, the expected loss-weighted tier influence is 51.0% from Tier 1, 33.9% from Tier 2, and 15.1% from Tier 3. These percentages describe an *expected* contribution under the sampling and weighting scheme, not the exact realised gradient share.

4.3.5 Per-Tier Data Augmentation

Data augmentation is applied after sampling and is adapted to the reliability of each tier. Cleaner crops are given stronger stochastic transformations, whereas noisier crops are kept closer to their original appearance.

Figure 4.2 shows the augmentation policies used for Tier 1 and Tier 2. Tier 1, which contains the cleanest curated crops, receives the strongest augmentation policy, including geometric, photometric, and occlusion-based transformations. Tier 2, composed of manually annotated student crops, uses a more moderate policy with milder geometric and photometric changes.

Tier 3 and validation are not shown in Figure 4.2 because both use deterministic preprocessing only. Tier 3 images are automatically mined video crops and random augmentation is disabled

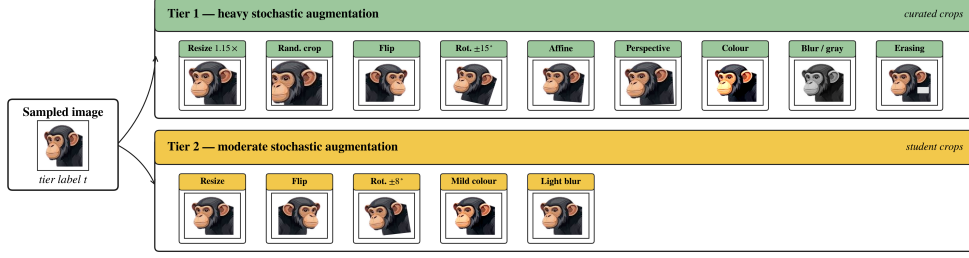


Figure 4.2: Per-tier augmentation policies used during training. Tier 1 receives the strongest stochastic augmentation, whereas Tier 2 receives a milder policy. (Chimpanzee image adapted from [28].)

in the final run using `-no-tier3-augmentation`. Both Tier 3 and validation images are resized to 224×224 , converted to tensors and normalised, no random augmentation is applied.

4.3.6 Evaluating the Impact of Tier 3 Data

To quantify the contribution of the automatically mined Tier 3 dataset, we conducted an ablation study in which the ChimpUFE model was fine-tuned *without* using any Tier 3 images.

Although the tier-aware sampling and loss weighting scheme is designed to mitigate tier 3 imbalance, we were interested to evaluate whether Tier 3 effectively improves the learned representation or whether its noise negatively impacts performance.

Experimental setup. The ablation model is trained under the exact same conditions as the ChimpUFE fine-tuned model with ArcFace, with the only difference being the complete exclusion of Tier 3 data from both sampling and optimisation. This is enforced by setting both the sampling multiplier and the loss weight of Tier 3 to zero. ($m_3 = 0, \lambda_3 = 0$).

For consistency with the main experiments, Tier 1 and Tier 2 are given equal importance in this ablation $m_1 = m_2 = 3.0, \lambda_1 = \lambda_2 = 1.0$.

The quantitative impact of this ablation is reported and analysed in the next Chapter section 5.2.4.

4.3.7 Validation Protocol

Validation is performed on the union of the Tier 1 and Tier 2 validation splits. Tier 3 is excluded because it is automatically mined and therefore less reliable than the manually curated tiers. Checkpoint selection is thus based only on clean identity annotations.

At the end of each epoch, the model is evaluated on this validation set. The checkpoint with the highest validation top-1 accuracy is retained. Training is considered converged when validation top-1 accuracy has not improved for 20 consecutive epochs.

4.4 Post-Processing Methods

Overview. Body Re-ID systems are primarily designed for human pedestrian data, where individuals can often be distinguished by clothing colour, accessories, and gait. Chimpanzees present a qualitatively different domain. Their fur texture and body silhouette are comparatively uniform across individuals, and the cues that are most discriminative for identity, such as facial scarring, ear shape, and supra-orbital ridges, are localised on the face. At the scale of our videos, these facial cues are often too small, blurred, or not visible inside body crops. Fine-tuning OSNet on chimpanzee bodies can make the embeddings better adapted to this domain, but it cannot recover identity information that is not present in the crop. In other words, if the face is absent or unreadable, a body-based Re-ID model can only rely on weak cues such as global shape or fur texture.

The face is the most reliable identity cue, but it is not visible often enough to drive online tracking. A tracker must associate detections at every frame, whereas usable face evidence may appear only intermittently. We therefore separate the task into two stages: ByteTrack follows bodies using motion cues, and an offline post-processing stage assigns identities by accumulating face evidence over each tracklet.

We implemented two versions of the post-processing layer. The first, developed before fine-tuning the ChimpUFE backbone, recovered identities by clustering per-tracklet face embeddings. The clustering was constrained by co-alive tracks and by a silhouette-based choice of the number of identities. The second version is the production auto-labelling pipeline. It assigns names with the ChimpUFE fine-tuned model and then improves temporal consistency through cluster merging, label propagation, per-frame deduplication, and gap interpolation. Results for both versions are reported in Chapter 5.

4.4.1 Dual-Heuristic Spatio-Temporal Post-Processing

Description

Setting. This first version uses the original ChimpUFE pretrained backbone, before the ArcFace identity head was available. Since the model could produce embeddings but not names, identity recovery was formulated as unsupervised clustering over ByteTrack tracklets. Each tracklet T_i had to be converted into one anonymous signature, and the goal was to partition the set of tracklets into consistent identities.

Step 1 – Smart subsampling. Rather than embedding every frame of a tracklet, we sample five representative crops uniformly over its lifespan and retain the most informative face views using the same YOLOX detector from chimpUFE [26] and the same frontal-face filtering conventions defined for Tier 3 (see section 3.5.5).

Step 2 - Fingerprinting with ChimpUFE. Each crop is passed through the original ChimpUFE backbone, which outputs a 768-dimensional embedding vector. Because ChimpUFE was

not yet fine-tuned with the identity head, these vectors should be read as anonymous appearance fingerprints rather than named predictions.

Step 3 - Creating a track signature. For a tracklet T_i , the five embeddings are averaged and ℓ_2 -normalised into a single vector e_i . This *track signature* compresses the full tracklet into one identity descriptor that is more stable than any individual frame.

Step 4 - The comparison matrix. Once every tracklet has a signature, we compare all pairs through cosine distance,

$$D_{ij} = 1 - e_i^\top e_j. \quad (4.4)$$

Small values indicate visually compatible tracklets, whereas large values indicate different chimpanzees. Running this comparison over all N tracklets yields an $N \times N$ distance matrix, illustrated in Figure 4.3. Low-distance blocks along the matrix reveal groups of tracklets that are likely to belong to the same individual.

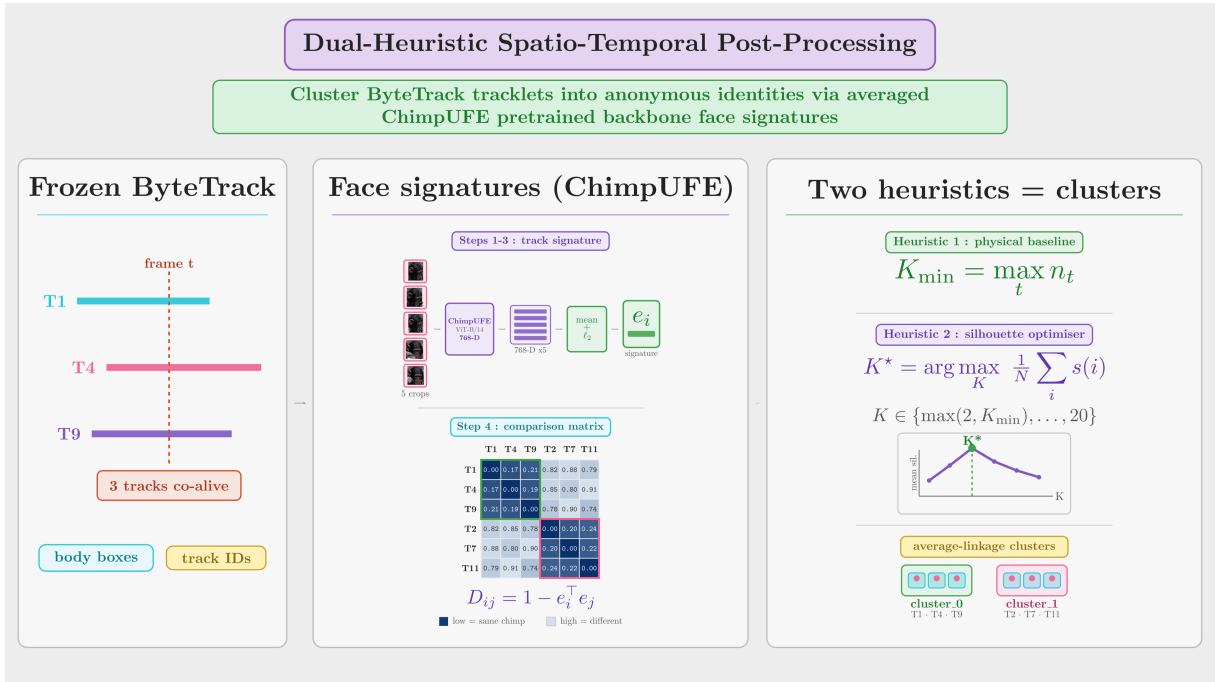


Figure 4.3: Dual-heuristic post-processing pipeline illustration.

Heuristic 1 - Physical baseline. A clustering of tracklets cannot use fewer identities than the maximum number of tracklets simultaneously alive in the same frame. If n_t denotes the number of active tracklets at frame t , then

$$K_{\min} = \max_t n_t. \quad (4.5)$$

This is the first heuristic: it enforces a physically plausible lower bound on the number of chimpanzees present in the video.

Heuristic 2 – Silhouette optimizer. We then test candidate cluster counts in the set

$$\mathcal{K} = \{\max(2, K_{\min}), \dots, 20\}$$

using average-linkage agglomerative clustering [29] on the precomputed distance matrix. For each candidate clustering, every tracklet i receives a silhouette score [30] $s(i)$. This score is high when the tracklet is close to other tracklets in its own cluster and far from tracklets in neighbouring clusters.

The final number of clusters is selected by maximising the mean silhouette over all N tracklets:

$$K^* = \arg \max_{K \in \{\max(2, K_{\min}), \dots, 20\}} \frac{1}{N} \sum_{i=1}^N s(i). \quad (4.6)$$

This second heuristic favours clusterings that are compact within clusters and well separated across clusters.

Agglomerative clustering. After selecting K^* , we run average-linkage agglomerative clustering with that value of K and assign one anonymous identity index to each resulting cluster. In practice this is where fragmented tracklets that share the same chimpanzee, such as tracks 1, 4, and 9 in a typical video, are merged into one stable identity group.

Limitations. This first iteration already improves identity consistency, but it has two structural limitations. First, the signature is built from only five representative crops and no explicit name classifier exists, so difficult poses or weak face visibility can still force visually similar chimpanzees to be over-merged. Second, the output remains anonymous (`cluster_0`, `cluster_1`, ...) rather than operationally useful chimpanzee names. These limits motivated the second iteration, described next, which replaces the anonymous clustering stage by an explicit face-recognition and label-propagation pipeline.

4.4.2 Face-Recognition-Driven Auto-Labeling Pipeline

Data for Face Re-Identification

The auto-labelling pipeline is built on the same face corpus used for fine-tuning the ChimpUFE pretrained backbone. The construction of this corpus, including the source videos, YOLOX-based crop mining, quality filtering, three-tier organisation, and identity distribution, was described in section 3.5.5. We therefore do not repeat these details here. All stages below assume the same 20 identities, face detector, and crop-quality conventions.

Description

Setting. The pipeline described here is the production system shipped with the current thesis. It supersedes the dual-heuristic post-processing in two essential ways. First, it uses the ChimpUFE fine-tuned model to produce explicit chimpanzee names rather than anonymous cluster indices. Second, it replaces the body-crop face-detection stage of the dual-heuristic pipeline by a

YOLOX face-in-frame cascade. Faces are detected once on the full frame and are then attributed to body tracks. The downstream consequence is that every body box published by the tracker can be tagged either with a known chimpanzee name, one of the 20 known identities, or with an `unknown_n` placeholder when the face evidence is insufficient.

Pipeline overview. The full pipeline has eleven stages. Stages 1–2 are online and produce cached tracker and face-crop outputs. Stages 3–11 are offline post-processing stages that classify, cluster, repair, and export the final labels. Figure 4.4 gives the global overview and the detailed figures that follow break the offline post-processing into three readable blocks.

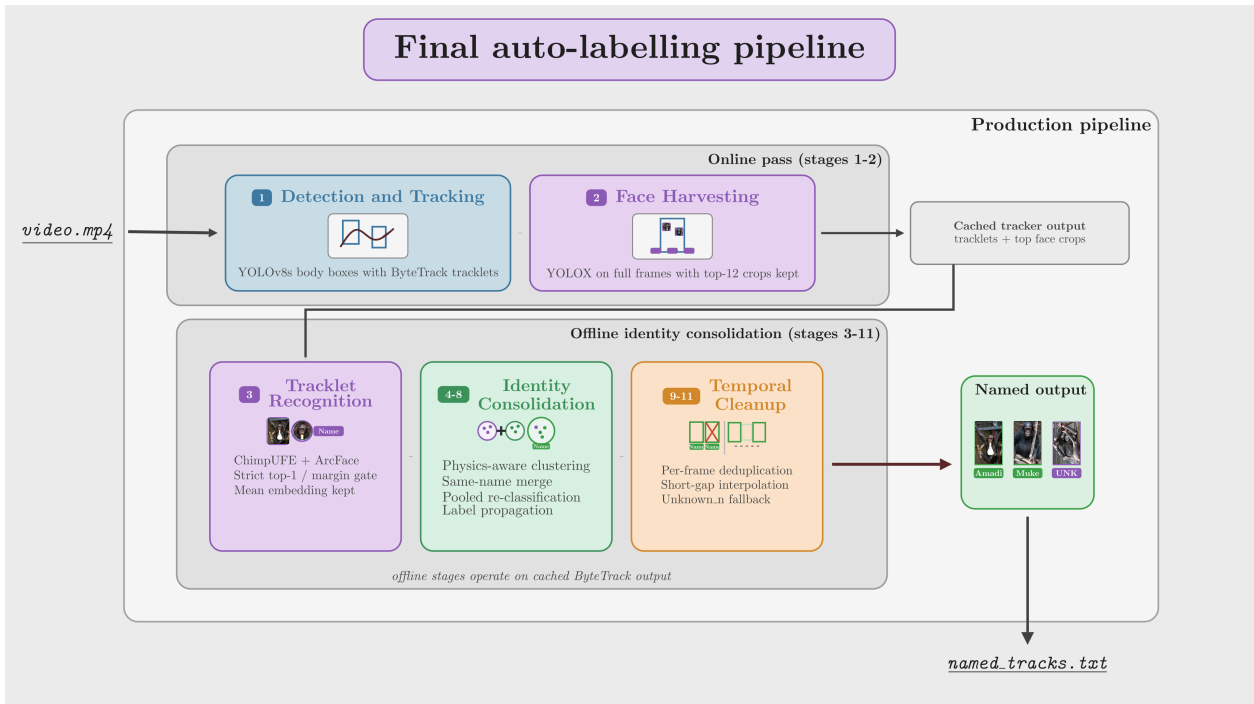


Figure 4.4: Global auto-labelling pipeline. Stages 1–2 are online. Stages 3–11 are offline post-processing on the cached tracker output.

Stages 1–2 - Online detection, tracking, and crop mining. The online part of the pipeline produces the raw material used by all later stages. In stage 1, the YOLOv8s body detector is combined with the ByteTrack configuration. The configuration table can be found in Appendix Table B.6. The output is a set of body tracklets.

$$\mathcal{T} = \{T_1, \dots, T_N\}.$$

Each tracklet is a sequence of body boxes over time. Tracklets shorter than 30 frames are discarded as transient noise.

In stage 2, faces are detected using the same pretrained YOLOX [26] face detector used to mine the Tier 3 corpus. Unlike the dual-heuristic pipeline, which ran face detection separately inside each cropped body region, the present pipeline applies YOLOX to the *full frame*. A

detected face is assigned to tracklet T_i at frame f if the centre of the face box lies inside the body box of T_i at f . A tracklet whose best detection face confidence is below 0.2 keeps no crop and enters the offline pipeline as an unlabelled candidate.

For each tracklet T_i , up to 60 frames are sampled uniformly. Face candidates assigned to the tracklet are filtered with a production crop-quality procedure related to the Tier 3 filter of section 3.5.5, but with the auto-labelling thresholds listed in Table C.1. Among the surviving candidates, each crop is sorted by score, and the top 12 crops are retained.

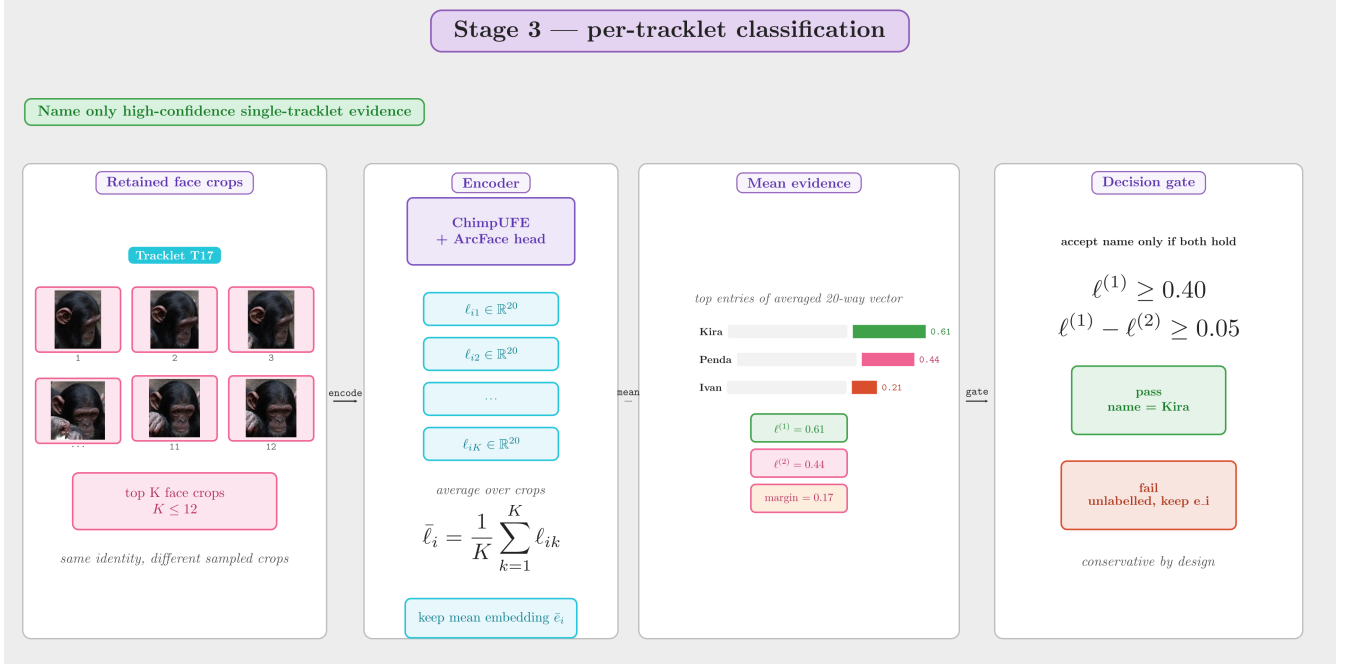


Figure 4.5: Stage 3 of the auto-labelling pipeline. Up to 12 retained face crops from a single tracklet are encoded with ChimpUFE and ArcFace. The averaged classifier evidence is accepted only if the top score and top-1 versus top-2 margin pass the strict early naming gate. Otherwise, the tracklet remains unlabelled but keeps its mean embedding for later clustering.

Stage 3 - Strict per-tracklet naming. Figure 4.5 shows the first offline decision point. Each retained face crop of a tracklet is forwarded through the fine-tuned ChimpUFE backbone and ArcFace head. For crop k of tracklet T_i , the model produces two related but distinct objects:

$$e_{ik} \quad \text{and} \quad \ell_{ik} \in \mathbb{R}^{20}.$$

Here, e_{ik} is the embedding vector of the crop, whereas ℓ_{ik} is the 20-dimensional class-score vector containing cosine similarities to the 20 identity prototypes. The tracklet-level classifier evidence is obtained by averaging the score vectors over the retained crops,

$$\bar{\ell}_i = \frac{1}{K} \sum_{k=1}^K \ell_{ik}, \quad K \leq 12.$$

The tracklet receives an explicit name only if the strongest class score and the margin to the

runner-up both pass the strict gate

$$\ell^{(1)} \geq \tau_{\text{cls}} = 0.40 \quad \text{and} \quad \ell^{(1)} - \ell^{(2)} \geq \delta_{\text{cls}} = 0.05. \quad (4.7)$$

Here, $\ell^{(1)}$ denotes the highest entry of the averaged score vector $\bar{\ell}_i$, and $\ell^{(2)}$ denotes the second-highest entry. The first condition requires the best identity hypothesis to be sufficiently strong and the second requires it to be separated from the closest alternative. Tracklets that fail this gate remain unlabelled, but they keep their mean embedding $\bar{e}_i \in \mathbb{S}^{767}$ for the clustering stages. The gate is deliberately conservative, at this point the classifier only sees evidence from a single tracklet, and an incorrect early name is difficult to undo later.

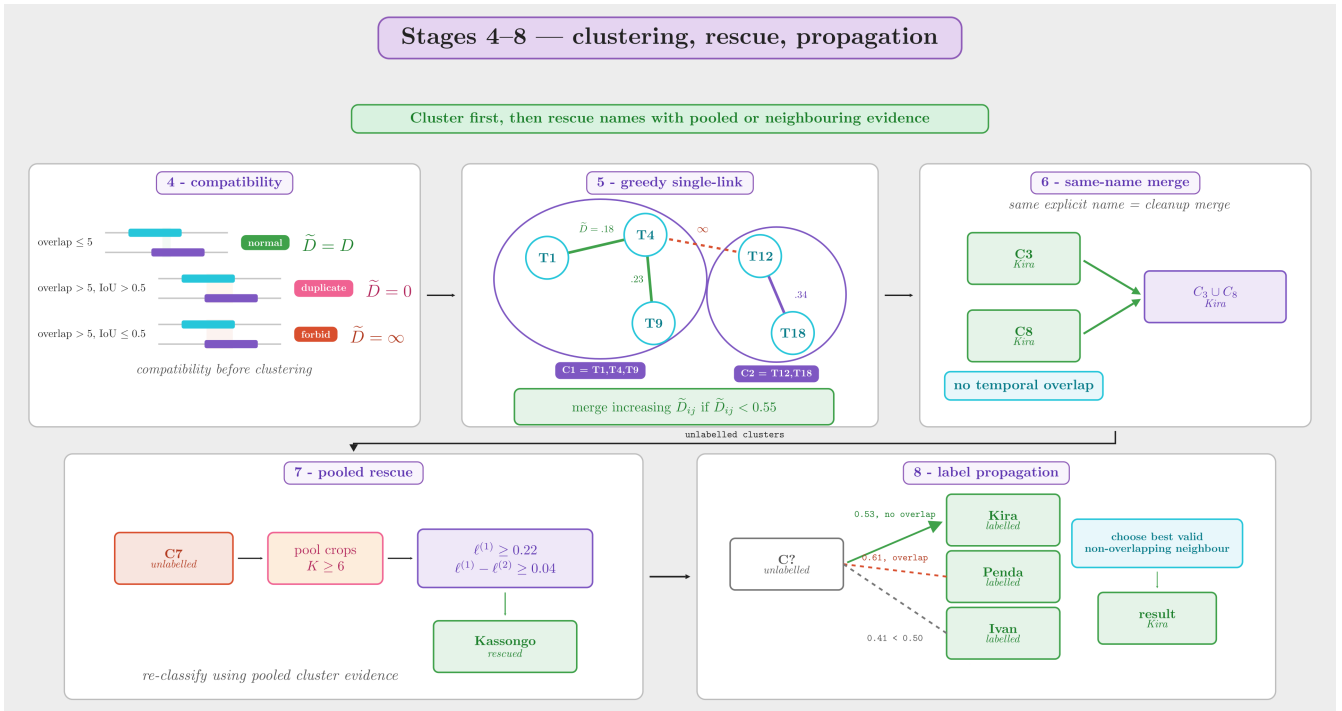


Figure 4.6: Stages 4–8 of the auto-labelling pipeline. Tracklets are first grouped using overlap-aware embedding distances. Same-name disjoint clusters are merged, unlabelled clusters are re-classified with pooled evidence, and remaining unlabelled clusters can inherit a label from their most similar non-overlapping labelled neighbour.

Stages 4–8 - From tracklets to labelled clusters. Figure 4.6 summarises the central clustering and rescue part of the pipeline. The goal of these stages is to turn isolated tracklets into identity-level clusters, while preventing physically impossible merges.

Stage 4 - Compatibility and physics penalty. The pipeline first computes a pairwise cosine distance between mean tracklet embeddings:

$$D_{ij} = 1 - \bar{e}_i^\top \bar{e}_j.$$

Since the embeddings are normalised, $\bar{e}_i^\top \bar{e}_j$ is the cosine similarity. A small value of D_{ij} therefore means that two tracklets look similar in embedding space. However, appearance alone is not

enough, two tracklets that are clearly alive at the same time in different places should not be merged. The raw distance D_{ij} is therefore converted into a modified distance $\tilde{D}_{ij}$.

As shown in the left panel of Figure 4.6, there are three cases. If two tracklets overlap for at most $\tau_{\text{ovl}} = 5$ frames, the normal distance is kept. If they overlap for more than 5 frames and their mean IoU exceeds $\tau_{\text{IoU}} = 0.5$, they are treated as phantom duplicates of the same body and are forced to merge by setting $\tilde{D}_{ij} = 0$. If they overlap for more than 5 frames but have low IoU, they are treated as genuinely co-alive bodies and the merge is forbidden by setting $\tilde{D}_{ij} = +\infty$.

Stage 5 - Greedy single-link cluster merge. The modified distances are then used for greedy single-link clustering. Tracklet pairs are processed in increasing order of $\tilde{D}_{ij}$, and every pair satisfying

$$\tilde{D}_{ij} < \tau_{\text{merge}} = 0.55$$

is merged. The physics penalty from stage 4 ensures that the greedy procedure can merge visually similar fragments while never merging two tracklets that are physically incompatible. Each resulting cluster aggregates the tracklets, crops, embeddings, and any names assigned at stage 3.

Stage 6 - Same-name cluster merge. The next cleanup step merges clusters that already carry the same explicit name, provided that they do not overlap in time beyond the overlap threshold. For example, if two disjoint clusters were independently and confidently named Kira at stage 3, they are merged into a single Kira cluster.

Stage 7 - Cluster-aggregated re-classification. Some clusters remain unlabelled after stages 4–6. These clusters are re-evaluated using pooled evidence from all their retained crops. A cluster must contain at least 6 retained crops to be considered. Because the decision now uses more evidence than a single tracklet, the classifier gate is relaxed:

$$\ell^{(1)} \geq \tau_{\text{cls}}^c = 0.22 \quad \text{and} \quad \ell^{(1)} - \ell^{(2)} \geq \delta_{\text{cls}}^c = 0.04. \quad (4.8)$$

This stage can rescue clusters that were too uncertain to name at the single-tracklet level but become reliable once their evidence is pooled. After this rescue step, stage 6 is applied again so that newly named clusters can be merged with already labelled clusters of the same identity.

Stage 8 - Embedding-based label propagation. Finally, clusters that remain unlabelled after stage 7 are compared to labelled clusters using embedding similarity. A label may be propagated only if the similarity exceeds

$$\tau_{\text{prop}} = 0.50$$

and the two clusters do not overlap in time. Each unlabelled cluster inherits the label of its most similar valid non-overlapping labelled neighbour. If no labelled neighbour passes the gate, the cluster remains unlabelled. This stage copies only the identity label, it does not alter the geometry of the boxes.

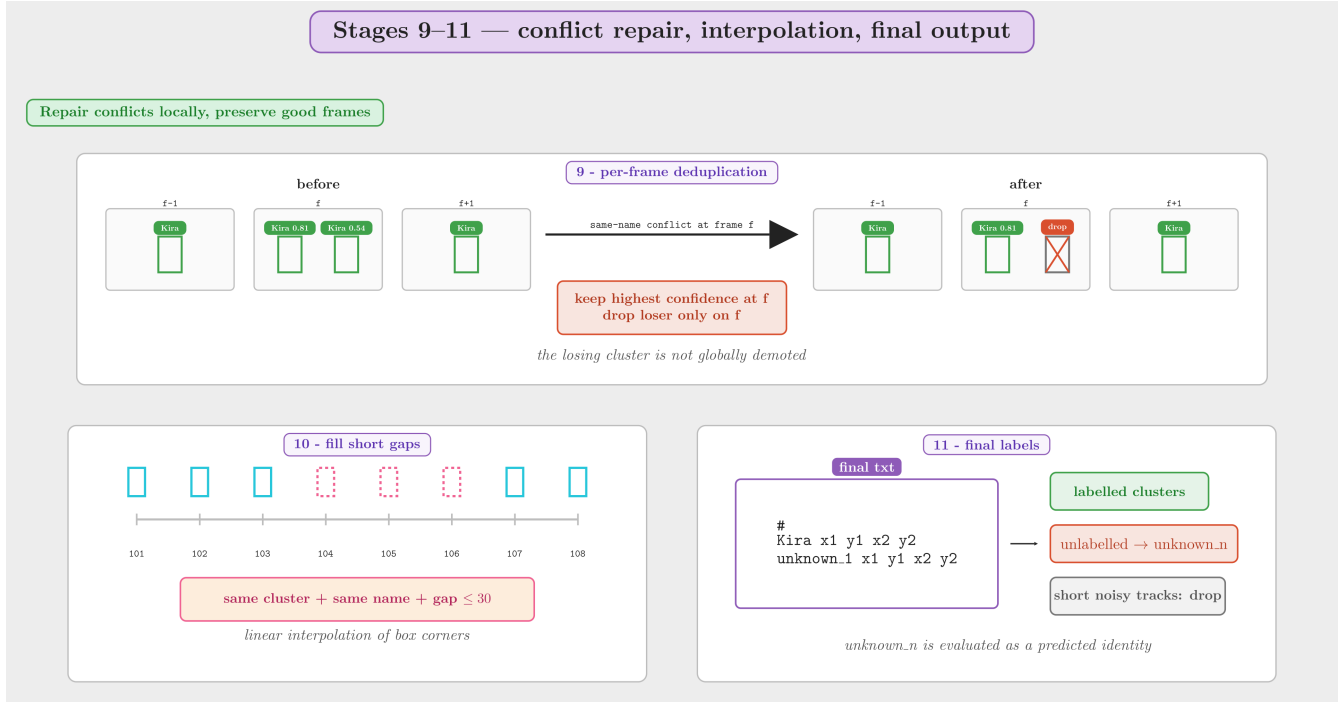


Figure 4.7: Stages 9–11 of the auto-labelling pipeline. Remaining same-name conflicts are repaired locally on the conflicting frames only, short gaps are interpolated for named clusters, and the final per-frame text output is produced with unresolved clusters assigned **unknown_n** identities.

Stages 9–11 - Conflict repair, interpolation, and final output. Figure 4.7 shows the final repair and export stages. At this point, most tracklets have either been assigned a name or grouped into an unlabelled cluster, but small conflicts and temporal gaps can still remain.

Stage 9 - Per-frame deduplication. After stages 5–8, two clusters with the same assigned name can still occasionally be co-alive on the same frame. Earlier versions of the pipeline resolved this by demoting the lower-confidence cluster to **unknown_n** entirely. This was too destructive, because the demoted cluster lost its name even on frames where there was no conflict. The final pipeline therefore performs a local per-frame repair. On a frame where the same name appears more than once, only the box with the highest confidence keeps the name. The lower-confidence duplicate is dropped on that frame only. On non-conflicting frames, the losing cluster keeps its name.

Stage 10 - Gap interpolation. For each named cluster, short temporal gaps between fragments carrying the same name are filled if the gap length is at most 30 frames:

$$\text{gap} \leq 30.$$

The missing boxes are obtained by linear interpolation of the bounding-box corners (x_1, y_1, x_2, y_2) . This recovers short detector drop-outs and brief occlusions that survived ByteTrack’s own temporal buffer.

Stage 11 - Output and unknown handling. The final output is a per-frame text file. Each frame begins with a # delimiter and is followed by one line per surviving box:

`<name> <x1> <y1> <x2> <y2>.`

Clusters that remain unlabelled after stage 8 are assigned enumerated placeholders `unknown_1`, `unknown_2`, and so on. These `unknown_n` identities are preserved during HOTA evaluation. They count as valid predicted identities, but they cannot match any named ground-truth chimpanzee.

Hyperparameters and version progression. The gates introduced above are collected in Table C.1 in the appendix. None of them was subjected to a Cartesian search of the size used for the trackers. They were set by hand, by inspecting the evolution of the labelled-tracklet count and the per-frame conflict count across the eleven-version development trail of Table C.2, also in the appendix. A grid search over these gates is left to future work Chapter 6.

Experiments

The auto-labelling pipeline is evaluated on the same evaluation videos (J, K) as the trackers. The body detector and ByteTrack are frozen at the configuration reported in Table B.6, and the ChimpUFE fine-tuned checkpoint is reused without modification. Only the post-processing stages vary between runs. The HOTA evaluation pipeline described in the previous chapter section 3.3 is reused verbatim, and the `unknown_n` convention is preserved. Numerical results are reported in section 5.4.1.

Chapter 5

Results and Analysis

Chapter Overview This chapter evaluates the complete progression from face-model training to end-to-end auto-labelling. It first reports the fine-tuning results of the ChimpUFE face embedder, including aggregate identification scores, per-identity behaviour, the contribution of Tier 3 data, embedding geometry, and confidence-based abstention. It then compares the on-line trackers on the tuning and evaluation protocols, with particular attention to the HOTA decomposition and to the curated nature of the ground truth. Finally, it evaluates the two post-processing stages on the evaluation videos J and K, analyses the identity-aware recognition output of the final pipeline, and discusses the remaining failure modes. The central question of the chapter is whether the thesis contributions improve the practical annotation pipeline by solving the association and naming problem, rather than merely changing detector behaviour.

5.1 Experimental Protocol and Metric Caveats

Face Models Terminology. In this thesis, *backbone* refers to the feature extractor that maps a face crop to an embedding vector. A *head* refers to a parametric classifier on top of embeddings (here, an ArcFace head). When a model has *no head*, it does not directly output an identity label, identities are then obtained by *nearest-neighbour retrieval* in a labelled gallery using cosine similarity (the predicted identity is the label of the most similar gallery embedding).

Throughout the Results chapter, we use the following model names consistently.

- *ChimpUFE pretrained backbone* denotes the publicly released ChimpUFE[26] encoder used without any fine-tuning and evaluated through nearest-neighbour retrieval in the Tier 1 gallery.
- *ChimpUFE fine-tuned model (ArcFace)* denotes the end-to-end fine-tuned model and can be evaluated both by retrieval (Rank- k /mAP) and by its ArcFace head (reported as *Head T1*).
- *ChimpUFE pretrained backbone + ArcFace head* denotes the Stage 1 model (frozen backbone, trained head) and is evaluated using head predictions.

- *FaceNet-FC baseline* denotes the inherited FaceNet-based model [31, 1] from the previous pipeline and is evaluated through the same retrieval protocol for fair comparison.

In tables and figures, we use the short labels **ChimpUFE-PT**, **ChimpUFE-FT**, and **FaceNet-FC** to refer to these three models.

5.2 Face Embedder Fine-Tuning Results

5.2.1 Training Dynamics

Figure 5.1 shows the per-class validation accuracy across epochs. At the very beginning of training, accuracy is low because the ArcFace classification head is newly initialised. However, many identities improve rapidly during the linear-probing stage, while the ChimpUFE backbone is still frozen. This indicates that the ChimpUFE pretrained backbone already provides a useful and partially discriminative representation of chimpanzee faces. Full fine-tuning then further specializes this representation, with a smaller subset of harder classes converging more slowly and it remaining less stable across epochs.

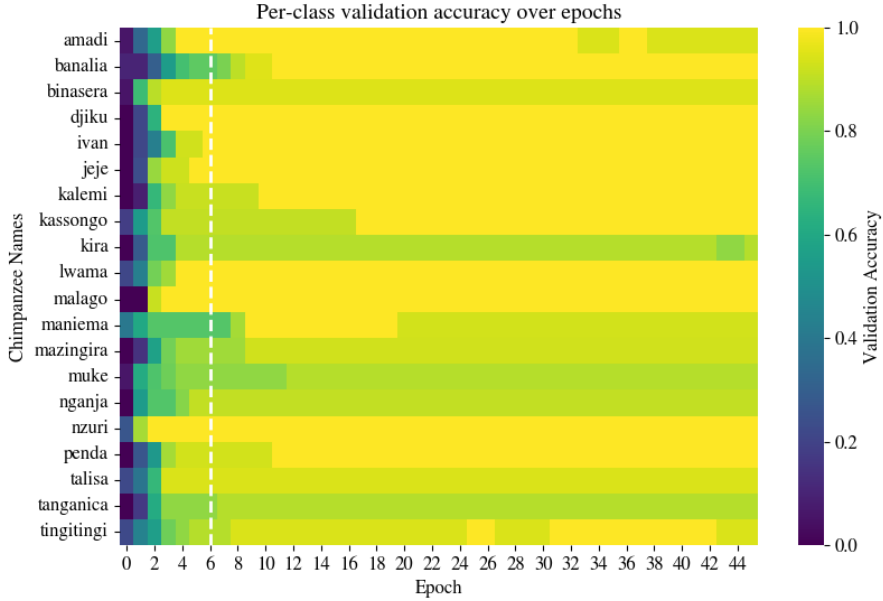


Figure 5.1: Per-class validation accuracy heatmap. The dashed line marks the stage transition (see section 4.3.3)

5.2.2 Test-Set Identification Performance

We compare three face embedders on the curated test set of section 3.5.6, which contains 500 face crops of the 20 chimpanzee identities of the study group, with 25 images per identity. The protocol is identical for every model. Each image is encoded into an ℓ_2 -normalised embedding. Rank- k and mAP are computed by comparing each test embedding to the train embeddings used as gallery. The train embeddings come from the training set of Tier 1. For embedding-only

models (FaceNet-FC baseline and the ChimpUFE pretrained backbone), identity is obtained by nearest-neighbour retrieval in the Tier 1 gallery using cosine similarity. The ChimpUFE fine-tuned model (ArcFace) additionally provides a parametric ArcFace head, reported as *Head T1*.

Table 5.1: Three face models compared on the curated test set. FaceNet-FC is the inherited baseline from last year, ChimpUFE-PT is the ChimpUFE pretrained backbone evaluated by nearest-neighbour retrieval, and ChimpUFE-FT is the ChimpUFE fine-tuned model (ArcFace).

Model	Backbone	Embedding dim.
FaceNet-FC	InceptionResNetV1	512
ChimpUFE-PT	ViT-B/14	768
ChimpUFE-FT(ours)	ViT-B/14	768

Table 5.2 reports the main identification metrics on the test set. Head T1 is the top-1 accuracy of the ArcFace classification head and is reported only for the ChimpUFE fine-tuned model (ArcFace), since the two embedding-only baselines are evaluated only through their embeddings. Rank- k and mAP are computed with the train set as gallery, following the Cumulative Matching Characteristic and mean Average Precision protocol introduced in section 2.4.1.

Table 5.2: Aggregate identification performance on the test set.

Model	ArcFace head Top-1	Rank-1	Rank-5	mAP
FaceNet-FC	–	0.072	0.120	0.127
ChimpUFE-PT	–	0.276	0.422	0.181
ChimpUFE-FT (ours)	0.868	0.854	0.870	0.830

The three models fall into three clearly separated regimes. The FaceNet-FC baseline performs close to chance on this test set and its predictions are largely unbalanced (see confusion matrix in Appendix E). It has a Rank-1 accuracy of only 7.2%. Its embeddings were learned on the smaller chimpanzee corpus available in the previous thesis and do not transfer reliably to the balanced 20-identity test set used here. The pretrained ChimpUFE backbone performs better, reaching 27.6% Rank-1, which confirms that its self-supervised pretraining already captures useful chimpanzee-face information. However, this level remains insufficient for reliable explicit identity assignment.

After the tier-aware fine-tuning of section 4.3.4, our ChimpUFE fine-tuned model reaches 85.4% Rank-1 on the test set. This corresponds to a +57.8% absolute gain in Rank-1 over the ChimpUFE pretrained backbone. The mAP also increases from 0.181 to 0.830, showing that the fine-tuned embeddings do not only retrieve the correct identity as the nearest match more often, but also rank multiple correct gallery images much higher. The ArcFace classification head reaches a comparable top-1 accuracy of 86.8%.

Figure 5.2 reports the full Cumulative Matching Characteristic up to Rank-20. The fine-tuned model dominates the two baselines across the entire rank range. The ChimpUFE pretrained backbone has lower performance than the fine-tuned model but still clearly outperforms the FaceNet-FC baseline.

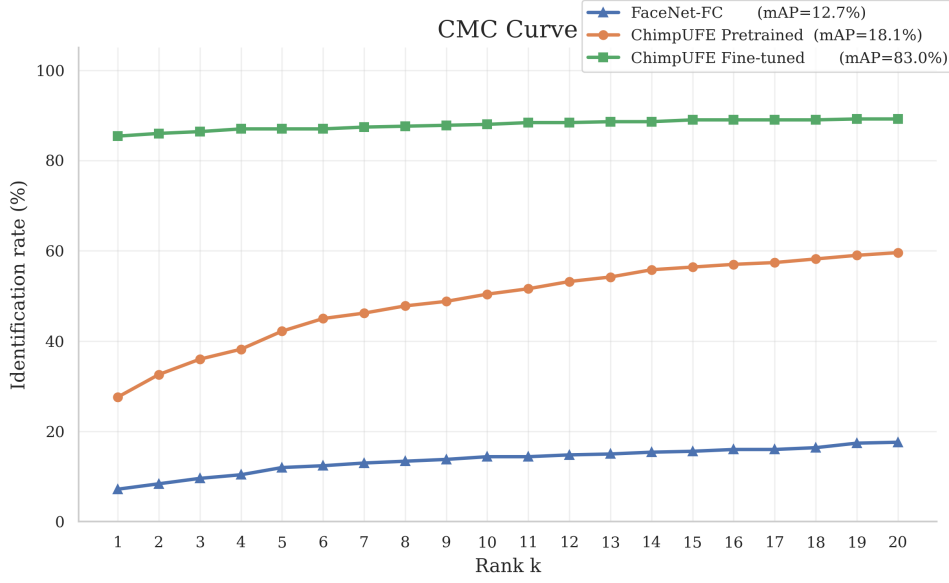


Figure 5.2: Cumulative Matching Characteristic on the test set, with the Tier 1 train set as gallery. ChimpUFE-FT dominates ChimpUFE-PT at every rank, and both outperform FaceNet-FC.

5.2.3 Per-Identity Behavior and Failure Modes

Figure 5.3 reports the per-identity F1-score of the fine-tuned classification head on the balanced face test set. Since each identity contributes 25 test images, differences between identities are not caused by unequal test support. Most identities reach high F1 scores. In particular, *tingitingi*, *kira*, *ivan*, and *kalemi* all reach F1 scores around or above 0.96. The lower end of the distribution is concentrated in a smaller set of identities, with *maniema*, *kassongo*, *lwama*, and *nzuri* forming the weakest classes.

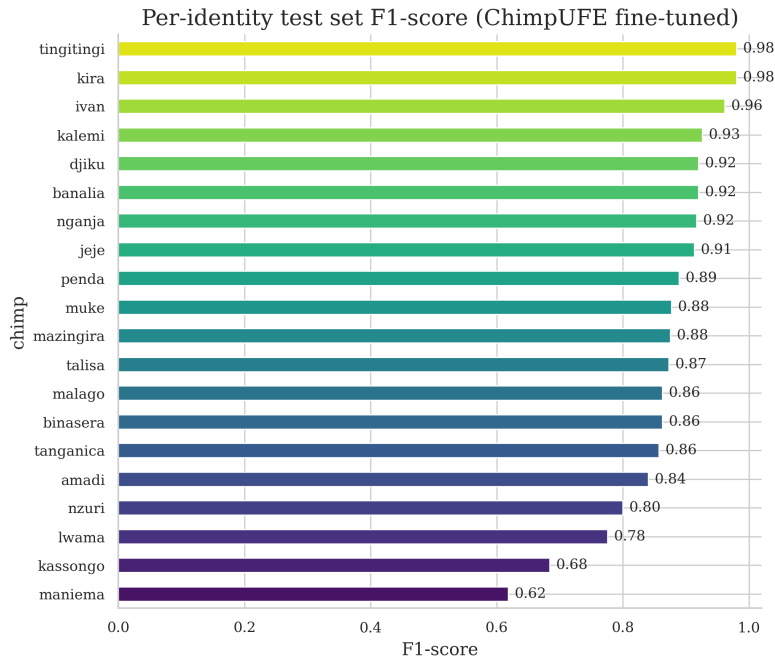


Figure 5.3: Per-identity F1-score on the face test set, using our fine-tuned model.

Additional confusion matrices for the three face models are reported in Appendix E.

The weakest identities do not fail in the same way. To make the comparison more informative, Table 5.3 contrasts the three lowest-F1 identities with the three highest-F1 identities, while also reporting their training support in each tier. The table also includes an F1 residual score. This residual is computed by first fitting a simple linear model that predicts per-identity F1-score from $\log(\text{total training images})$. The residual is then defined as

$$\text{Residual} = \text{F1}_{\text{observed}} - \text{F1}_{\text{predicted}}. \quad (5.1)$$

A negative residual means that an identity obtains a lower F1-score than expected from its amount of training data, whereas a positive residual means that it performs better than expected. This residual is only a diagnostic tool, it does not prove a causal effect of data volume, but it helps identify identities whose difficulty is not well explained by training-set size alone.

Kassongo is the clearest asymmetric case. Its precision is 1.000, but its recall is only 0.520. This means that predictions labeled as *kassongo* are correct, but many true *kassongo* crops are assigned to other identities. *Maniema* follows a different pattern, with both lower precision (0.567) and lower recall (0.680). Because F1 combines both precision and recall, *maniema* obtains the lowest F1-score and the strongest negative F1 residual in the table. *Lwama* is also below the expected trend, but less strongly. By contrast, the three high-performing identities have positive F1 residuals, meaning that they perform better than predicted from their total training support.

Table 5.3: Diagnostic comparison between low- and high-performing identities. Tier columns report the number of training images in each tier. The F1 residual is computed as the observed F1-score minus the F1-score predicted from $\log(\text{total training images})$. Negative values indicate identities that perform below the trend expected from data volume alone, while positive values indicate identities that perform above it.

Identity	Tier 1	Tier 2	Tier 3	Total	Precision	Recall	F1	F1 Residual
<i>Lowest F1 identities</i>								
maniema	44	15	354	413	0.567	0.680	0.618	-0.237
kassongo	31	20	579	630	1.000	0.520	0.684	-0.181
lwama	49	7	313	369	0.792	0.760	0.776	-0.077
<i>Highest F1 identities</i>								
kira	43	42	959	1044	0.962	1.000	0.980	0.104
tingitingi	38	42	1778	1858	0.962	1.000	0.980	0.091
ivan	51	2	353	406	0.926	1.000	0.962	0.106

The comparison in Table 5.3 does not support a purely data-volume explanation of the failure cases. *Kassongo* has 630 training images in total, including 579 Tier 3 crops, but remains among the weakest identities. Conversely, *ivan* reaches an F1-score of 0.962 with only 406 total training images, fewer than *kassongo* and several other lower-performing identities. Similarly, *kira* and *tingitingi* reach the highest F1-scores, and their positive residuals indicate that their performance is above what is predicted from data volume alone. These contrasts suggest that training-set size contributes to performance but is not sufficient to explain the class-level differences.

5.2.4 Effect of Tier 3 Data and Training-Set Size

The results above use the full tier-aware fine-tuning corpus, including the automatically mined Tier 3 face crops. To verify that these additional crops improve the model rather than simply adding label noise, we trained a second model under the same fine-tuning protocol but with Tier 3 disabled. In this ablation, only Tier 1 and Tier 2 contribute to sampling and to the loss. Table 5.4 reports the main test-set metrics for the two fine-tuned variants. Adding Tier 3 improves all reported metrics.

Table 5.4: Ablation on the contribution of Tier 3 data. Both models use the same ChimpUFE backbone and the same two-stage fine-tuning protocol. The only difference is that the automatically mined Tier 3 crops are included.

Model	Head T1	Rank-1	Rank-5	mAP
ChimpUFE-FT without Tier 3	0.762	0.620	0.738	0.521
ChimpUFE-FT with Tier 3	0.868	0.854	0.870	0.830

This ablation is consistent with the intended role of Tier 3 in section 4.3.4. Although Tier 3 images are noisier than the manually curated tiers, they also expose the model to a wider range of poses, illumination conditions, backgrounds, and video-quality artefacts. When their contribution is controlled through the tier-aware sampler and the low Tier 3 loss weight, this additional diversity appears beneficial for generalisation on the test set.

A complementary training-size diagnostic based on Spearman correlations suggests that training support is more associated with recall than with precision, and the full table and discussion are reported in Appendix E.2.

The per-identity analysis gives a more nuanced view of the same effect. Individual counterexamples remain. In particular, *kassongo* has substantial Tier 3 support but remains among the weakest identities, whereas *ivan* reaches high F1 with fewer total training images than several lower-performing identities. The global ablation therefore supports the use of Tier 3, while the per-identity diagnostics suggest that further improvements for the hardest identities may require targeted data enrichment or improved handling of identity-specific ambiguities, rather than simply increasing the number of mined crops.

For this reason, all subsequent analyses and the final auto-labelling pipeline use the full model trained with Tier 3.

5.2.5 Embedding Geometry

The identification scores of Table 5.2 depend on how well the embedding space separates images of different individuals while keeping images of the same individual close together. Table 5.5 reports, for the three models, the mean cosine similarity within each identity (intra), between distinct identities (inter), and the separation gap between the two. The fine-tuned embeddings show both higher intra-class similarity and lower inter-class similarity, meaning that images of the same chimpanzee form tighter clusters while images of different chimpanzees are pushed further apart.

Table 5.5: Embedding geometry on the test set. Intra and inter denote the mean cosine similarities of same-identity and different-identity pairs, respectively. Separation is computed as Intra – Inter, so larger values indicate better identity separation.

Model	Intra	Inter	Separation
FaceNet-FC	0.429	0.357	0.072
ChimpUFE-PT	0.357	0.249	0.108
ChimpUFE-FT (ours)	0.440	0.094	0.346

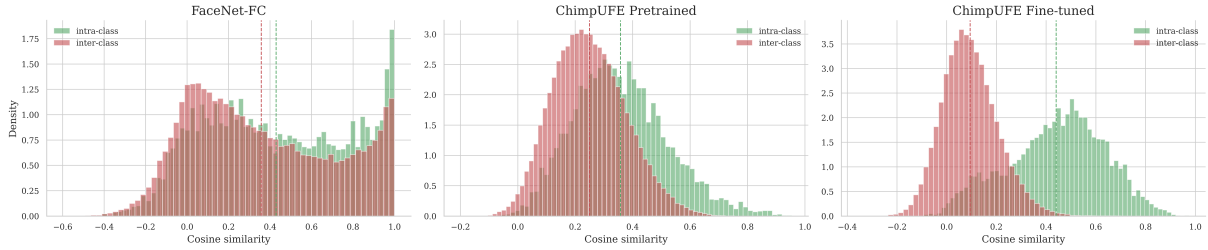


Figure 5.4: Distribution of pairwise cosine similarities for intra-class (same chimp, green) and inter-class (different chimp, red) pairs across the three evaluated models. The dashed vertical lines indicate the mean similarity for each distribution.

The same effect is visible qualitatively in Figure 5.5, which projects the test embeddings of the pretrained and fine-tuned backbones into two dimensions with t-SNE [32]. This projection is used only as a visual diagnostic as the axes have no physical unit and only the relative neighbourhood structure should be interpreted. In the pretrained embedding space, identities are less consistently separated and several colours overlap. In the fine-tuned model the embedding space becomes more identity-aware: many identities form more compact and better separated groups, although a few bridges remain between the hardest identities. This qualitative pattern is consistent with the quantitative gains in Rank-1 and mAP reported in Table 5.2.

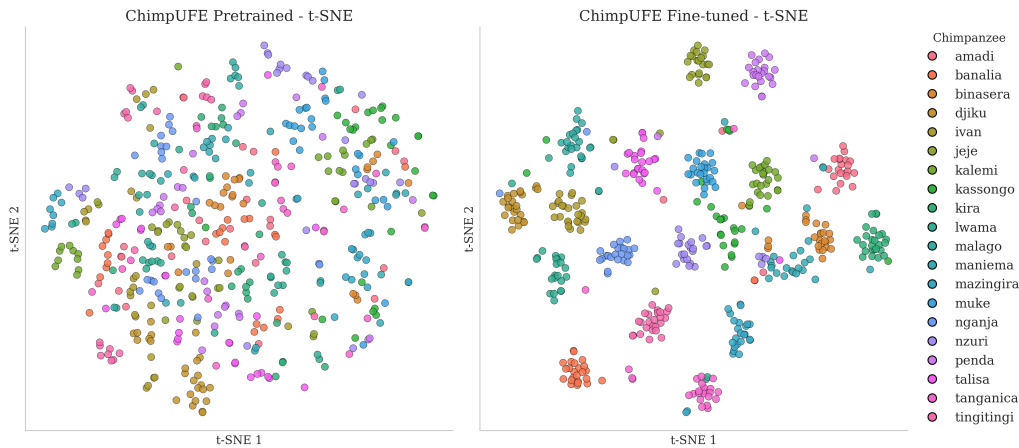


Figure 5.5: t-SNE projection of test embeddings for ChimpUFE-PT and ChimpUFE-FT. Each colour corresponds to one identity. The axes are arbitrary projection dimensions and have no physical unit. the figure is used only to visualise relative cluster structure.

5.2.6 Confidence-Based Abstention and Operating Points

The goal is not only to measure whether the model is accurate when forced to classify every image, but also whether its most confident predictions are more reliable, following the selective-classification intuition behind confidence-coverage analysis [33]. This is important for our post-processing. We define confidence using the gap between the two closest identity prototype embeddings. For a test embedding z , let s_1 be the cosine similarity to the closest prototype and s_2 the similarity to the second-closest prototype. The confidence margin is

$$\Delta = s_1 - s_2. \quad (5.2)$$

A large margin means that the predicted identity is clearly preferred over the nearest alternative, whereas a small margin indicates ambiguity between two possible identities. For example, if the closest prototype has similarity 0.70 and the second-closest has similarity 0.50, then $\Delta = 0.20$.

Figure 5.6 shows that this margin is informative. In the left panel, correct predictions are shifted toward larger margins, while wrong predictions are more concentrated near small margins. This means that many errors occur when the model hesitates between two plausible identities. The dashed thresholds illustrate how the margin can be used as an abstention rule. At $\tau = 0.2$, the model accepts 334 of the 500 test images. Among these accepted predictions, 321 are correct and 13 are wrong, corresponding to an accepted accuracy of 0.961. At the stricter threshold $\tau = 0.4$, only 137 images are accepted, but 134 of them are correct and only 3 are wrong, raising the accepted accuracy to 0.978.

The right panel shows the same behaviour as a confidence-coverage trade-off. Increasing the threshold reduces coverage, because more ambiguous images are rejected, but the accuracy on the retained subset increases. This is useful for the downstream auto-labelling pipeline, high-margin predictions can be used as safer identity seeds, while low-margin predictions can be left unlabelled or resolved later through tracklet-level evidence.

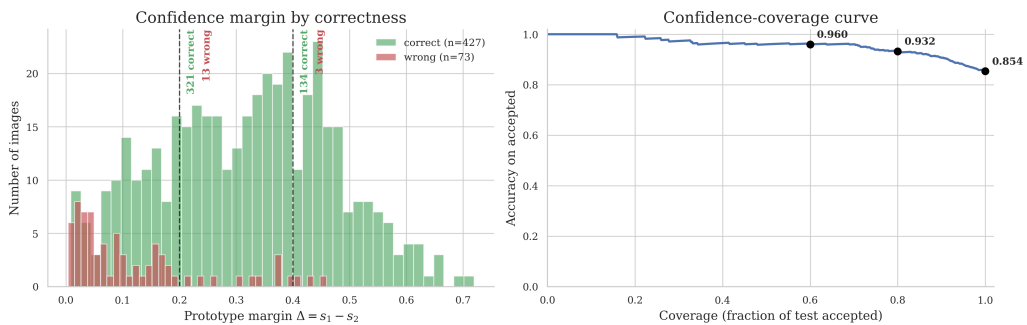


Figure 5.6: Confidence analysis of the fine-tuned model on the curated test set. Left: distribution of the prototype cosine margin $\Delta = s_1 - s_2$ for correct and incorrect predictions. The dashed thresholds show two possible abstention operating points. Right: accuracy on the accepted subset as a function of coverage when thresholding the same margin. Increasing the threshold rejects more ambiguous predictions, reducing coverage but improving reliability.

5.3 Tracker Methods Comparison

5.3.1 Evaluation Protocol and Ground-Truth Caveat

This section evaluates the online tracking stage before any face-based post-processing is applied. Its purpose is to choose the strongest upstream tracker and to test whether body appearance can improve frame-level association on this footage. The trackers therefore still output anonymous trajectory IDs, not chimpanzee names.

The comparison follows the split defined in section 3.2.2. StrongSORT, BoT-SORT, and ByteTrack are tuned on videos A (07h56, 14 individuals) and E (14h29, 3 individuals), then frozen and evaluated on videos J (12h33) and K (11h33). The inherited DeepSORT baseline is not tuned and is evaluated with its shipped configuration. Videos J and K were annotated after the tracker searches had concluded and are excluded from the OSNet body Re-ID training corpus and from Tier 3 face mining. Apart from their use in the standalone face-model test set, they are not used for training, validation, tracker tuning, gate selection, or checkpoint selection, and therefore test generalisation of the complete pipeline rather than performance on data seen during tuning.

One caveat is important before interpreting detection precision. The ground-truth files of section 3.2 are curated annotations, not exhaustive records of every visible chimpanzee. During manual correction, boxes that were too ambiguous, barely visible, or not useful for behavioural analysis were removed. As illustrated in Figure 5.7, the automatic pipeline can therefore keep boxes that are visually plausible but absent from the curated ground truth. These boxes are counted as false positives by our HOTA calculation mechanism, especially in dense scenes such as Video K.

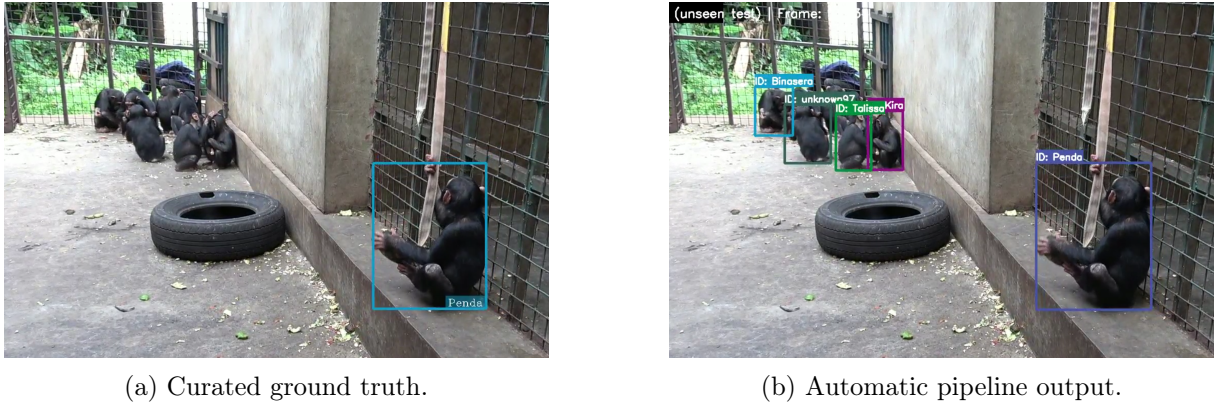


Figure 5.7: Crowded frame from Video K (11h33). The automatic pipeline retains every detection that survives the tracker thresholds. The curated ground truth keeps only the boxes the annotator considered useful for behavioural interpretation. Several visually plausible boxes in the right-hand image are therefore counted as false positives by HOTA.

5.3.2 Hyperparameter Tuning on the Tuning Videos

The tuning stage selects one frozen configuration per tracker before any evaluation-video comparison is made. The selected values are reported in Tables B.4 to B.6, the full grids, runtime

bookkeeping, and diagnostic plots are kept in Appendix B.

Three tendencies emerge from the searches. First, all selected configurations use a detection confidence of 0.5, preserving most real detections while suppressing the most ambiguous boxes. Second, the motion-driven trackers prefer strict IoU matching, ByteTrack performs best around `match_thresh = 0.9` and `track_high_thresh = 0.6` (Figure B.3b). Third, the tuned appearance terms remain secondary. StrongSORT obtains its best tuning behaviour when `mc_lambda` gives almost all weight to motion (Figure B.1b), and BoT-SORT is tuned with ByteTrack’s motion parameters frozen so that the Re-ID branch can be isolated cleanly.

5.3.3 Comparison on the Evaluation Videos

With those configurations frozen, Table 5.6 reports the full HOTA decomposition on the evaluation pair. This table answers the two following questions: which online tracker should feed the post-processing stage, and whether body appearance is strong enough to solve association before face evidence is introduced.

Table 5.6: Online tracker comparison on the evaluation videos. All trackers use the same YOLOv8s[1] body detector. Hyperparameters are frozen at the optimum found on Videos A and E (Tables B.4 to B.6). DeepSORT is reported with its shipped configuration without tuning.

Video	Tracker	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
J (12h33)	DeepSORT (inherited)	0.282	0.803	0.099	0.995	0.806	0.101	0.885
	StrongSORT	0.249	0.718	0.086	0.927	0.747	0.087	0.908
	BoT-SORT (motion only)	0.308	0.762	0.125	0.906	0.809	0.128	0.857
	BoT-SORT + chimp Re-ID	0.317	0.762	0.132	0.906	0.810	0.135	0.868
	ByteTrack	0.331	0.813	0.135	0.948	0.850	0.137	0.903
K (11h33)	DeepSORT (inherited)	0.266	0.551	0.129	0.994	0.553	0.133	0.879
	StrongSORT	0.242	0.498	0.118	0.943	0.509	0.121	0.909
	BoT-SORT (motion only)	0.287	0.532	0.156	0.926	0.551	0.169	0.832
	BoT-SORT + chimp Re-ID	0.293	0.532	0.161	0.926	0.551	0.171	0.858
	ByteTrack	0.298	0.560	0.159	0.961	0.573	0.172	0.852

ByteTrack is the most reliable upstream choice. Across the evaluation pair, ByteTrack gives the cleanest online starting point while remaining simpler than the appearance-augmented alternatives. The important result is not a large separation between trackers, but the opposite. The best online methods cluster close together and all leave the same association problem unresolved. This observation makes ByteTrack a pragmatic baseline for the rest of the chapter rather than the main source of the final improvement.

Body appearance does not solve association. Adding chimp-fine-tuned OSNet features to BoT-SORT changes the result only marginally, and StrongSORT remains weaker than the simpler motion-driven alternatives. This is consistent with the body Re-ID validation curves of Figure 5.8. The best OSNet checkpoint reaches only 34.5% Rank-1 and 22.1% mAP. Body embeddings are therefore above chance, but not discriminative enough to carry online association

in this setting. The embeddings are informative, but not discriminative enough to act as a reliable frame-level identity gate on crowded chimpanzee footage.

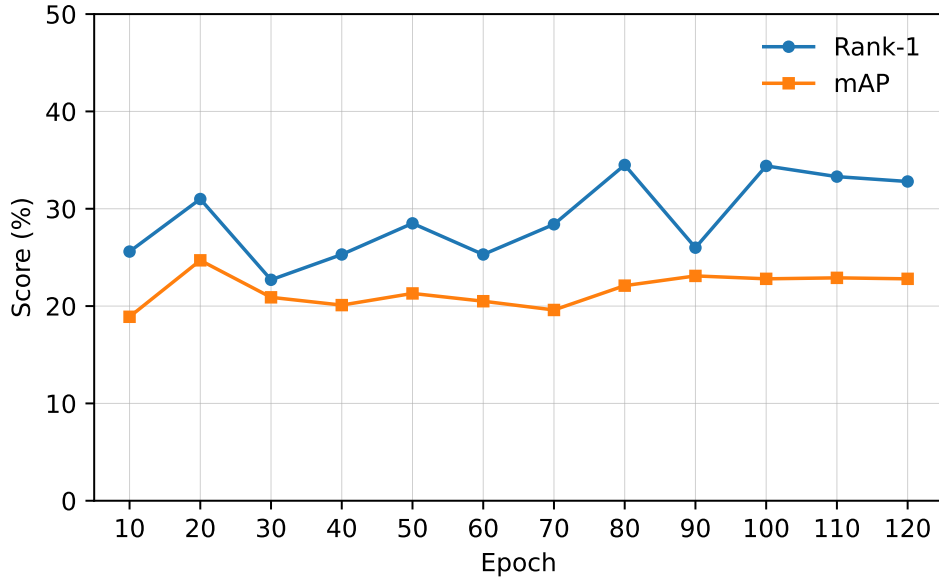


Figure 5.8: OSNet x1_0 training curves on the chimpanzee body-crop corpus (see section 3.4), fine-tuned for 120 epochs on 37 134 crops across 18 identities.

Detection is already strong, association is the bottleneck. The table is dominated by the same pattern for every tracker, detection is substantially stronger than association. The online stage usually finds the chimpanzees, but it cannot maintain stable identities through crossings, occlusions, and re-detections. The imbalance that motivated post-processing is therefore not an artefact of the tuning videos as it persists on the evaluation pair.

Video K is harder than Video J on the detection axis. Every tracker shows a clear detection-accuracy drop when moving from J to K, while association stays in the same range. Video K is filmed under denser crowding, and the gap between detection recall and detection precision indicates that YOLOv8s is not mainly missing chimpanzees. It emits boxes that the curated ground truth does not contain. This is the precision channel introduced earlier.

ByteTrack is therefore selected as the upstream tracker for all post-processing experiments. The conclusion is not that motion modelling is inherently superior to appearance modelling, but that body-only appearance is too weak at frame level on this footage. The remaining headroom must come from a stage that operates at the tracklet level and can integrate face evidence over time.

5.4 Post-Processing Results

Overview. The previous section shows that online tracking alone leaves association as the dominant failure mode. This section evaluates the two offline stages designed to address that gap. Both start from the same frozen ByteTrack output selected above and operate only at

the tracklet level. The dual-heuristic stage clusters anonymous fragments, while the final auto-labelling stage uses the fine-tuned ChimpUFE–ArcFace model to assign explicit chimpanzee names.

5.4.1 Post-Processing on the Evaluation Videos

With the pipeline frozen at the tuning-set configurations above, Table 5.7 reports the full HOTA decomposition for the same three stages on the two evaluation videos (J, K) that were never seen at any training or tuning step.

Table 5.7: End-to-end pipeline comparison on the evaluation videos J (12h33) and K (11h33). *Dual-heuristic* is the unsupervised face-clustering post-processor of section 4.4.1. *Final* is the auto-labelling pipeline of section 4.4.2. The ByteTrack row is repeated from Table 5.6 for easier reading.

Video	Stage	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr
J (12h33)	ByteTrack	0.331	0.813	0.135	0.948	0.850	0.137	0.903
	+ Dual-heuristic	0.382	0.815	0.179	0.949	0.851	0.507	0.246
	+ Auto-labelling (final)	0.628	0.799	0.493	0.920	0.858	0.567	0.787
K (11h33)	ByteTrack	0.298	0.560	0.159	0.961	0.573	0.172	0.852
	+ Dual-heuristic	0.307	0.562	0.167	0.963	0.574	0.684	0.205
	+ Auto-labelling (final)	0.521	0.544	0.498	0.921	0.570	0.644	0.717

Dual-heuristic stage. The dual-heuristic pipeline relies on the *unfine-tuned* ChimpUFE backbone and on the two heuristics that set a physical lower bound on K (peak concurrency) and then choose the final K by silhouette optimisation over the cosine-distance matrix. On both evaluation videos it produces only modest HOTA gains. Detection metrics barely move, which is expected because the dual-heuristic stage only relabels and stitches existing tracklets.

The most revealing pair of columns is (AssRe, AssPr). The dual-heuristic raises association recall sharply, but it pays for that gain with a severe collapse in association precision. In practical terms, the unsupervised clustering eagerly stitches fragments together, but it also over-merges different chimpanzees into the same cluster. Without an identity classifier the pipeline cannot distinguish a legitimate merge from a confusion, so the net HOTA gain remains limited. This is precisely the limitation that motivated the production pipeline.

Auto-labelling stage (final). The auto-labelling pipeline of section 4.4.2 replaces the anonymous clustering by an ArcFace classifier and a per-frame deduplication step. On the evaluation pair it yields the first large jump in HOTA. The dominant effect is not detection but the recovery of association precision without giving back most of the recall gained by the dual-heuristic stage. DetA changes only marginally, because the per-frame deduplication removes a small number of duplicate boxes but does not alter the detector itself. The final stage therefore succeeds for the reason the dual-heuristic stage failed, the face classifier can validate merges that the unsupervised clustering could only guess.

Figure 5.9 visualises the same table in bar form and makes the central point of this section

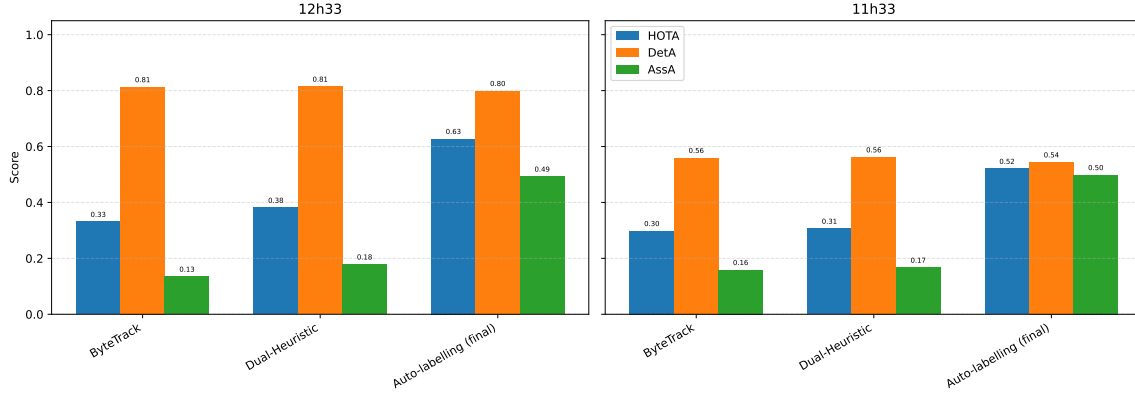


Figure 5.9: HOTA, DetA, and AssA breakdown on evaluation videos J (12h33) and K (11h33) for ByteTrack, the dual-heuristic post-processor, and the final auto-labelling pipeline. DetA remains nearly flat across the three stages, while the large HOTA gain appears only once AssA rises in the final stage.

explicit. Across the three post-processing stages, the decisive change is in AssA, not in DetA. ByteTrack, the dual-heuristic stage, and the final auto-labelling stage share nearly the same detection accuracy on each video, while the final stage produces the only large association gain. The pipeline therefore wins because it solves association, not because it discovers new detections.

The AssPr/AssRe pair of Table 5.7 exposes the same qualitative difference in a more compact way. The dual-heuristic stage trades precision for recall, whereas the final pipeline recovers both. The HOTA improvements measured on the evaluation videos confirm that the pipeline mechanism generalises to footage the system has never seen at any training or tuning step.

5.4.2 Recognition Accuracy of the Final Pipeline

HOTA scores detection and association quality, but after identity remapping it is intentionally insensitive to the literal name attached to a trajectory. A trajectory can therefore be temporally coherent while still carrying the wrong chimpanzee name. For this reason, the final auto-labelling pipeline is also evaluated with the identity-aware metrics defined in section 3.3.3 (IDAcc, IDRec, and the unknown rate).

Table 5.8 reports the identity-aware metrics on the evaluation pair. The pattern is consistent with the HOTA tables where most matched detections receive the correct chimpanzee name, Video K is slightly stronger than Video J on this metric, and the unknown rate remains secondary rather than dominating the error budget. The remaining gap is therefore driven more by genuine confusions between similar individuals than by abstention.

Table 5.8: Recognition accuracy of the final auto-labelling pipeline on the evaluation pair. Values are reported as scalar fractions for readability. The unknown-rate column is a lower-bound estimate obtained by aggregating the `unknown_n` entries that appear in the top-5 confusion print of the recognition step.

Video	IDAcc	IDRec	Unknown rate
J (12h33)	0.700	0.657	≥ 0.122
K (11h33)	0.781	0.733	≥ 0.119

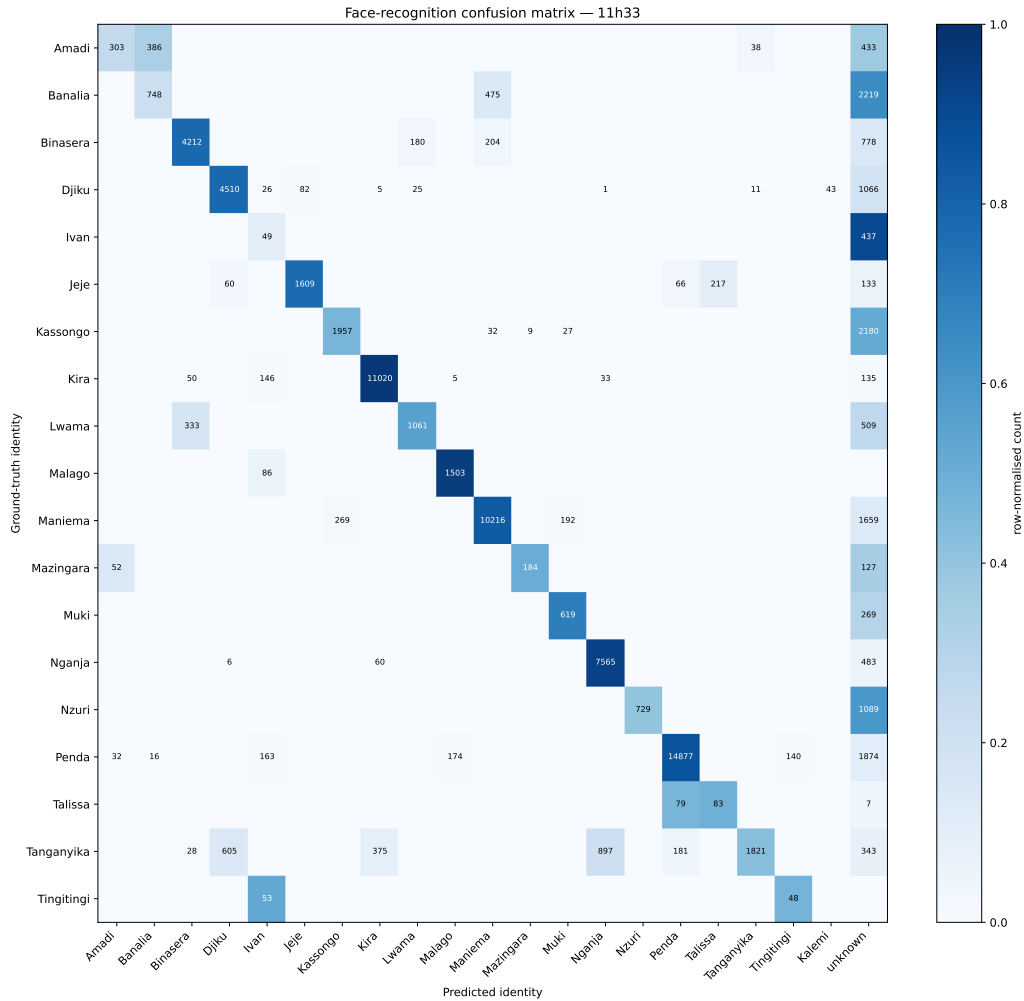


Figure 5.10: Identity confusion matrix of the final auto-labelling pipeline on Video K (11h33), one of the evaluation videos. The residual confusion concentrates in a small subset of visually similar individuals.

The corresponding confusion matrix helps explain the aggregate IDAcc and IDRec values. Most identities remain concentrated on the diagonal, while the residual off-diagonal mass is carried by a small subset of individuals. The per-identity breakdown shows that this is not simply the same ranking observed on isolated face crops. Some chimpanzees that are recognisable in curated face images become difficult in the full video because they appear farther from the camera, more often in profile, or in close contact with other individuals. The remaining recognition error is therefore split between two mechanisms, the intrinsic discriminative ceiling of the face model for visually similar individuals, and the visibility conditions imposed by the evaluation video itself.

Qualitative comparison. To make the difference between an anonymous tracker output and the final, identity-aware annotation tangible, two short fragments of Video K were rendered and uploaded as supplementary material. Figure 5.11 shows a matched frame from each fragment. On the left, the raw ByteTrack output displays only numeric track identifiers that change whenever a tracklet is broken and on the right, the auto-labelling pipeline replaces those identifiers with the chimpanzee names predicted by the ArcFace head, so that an identity that survives an occlusion

or a re-detection event keeps the same label across the entire clip.

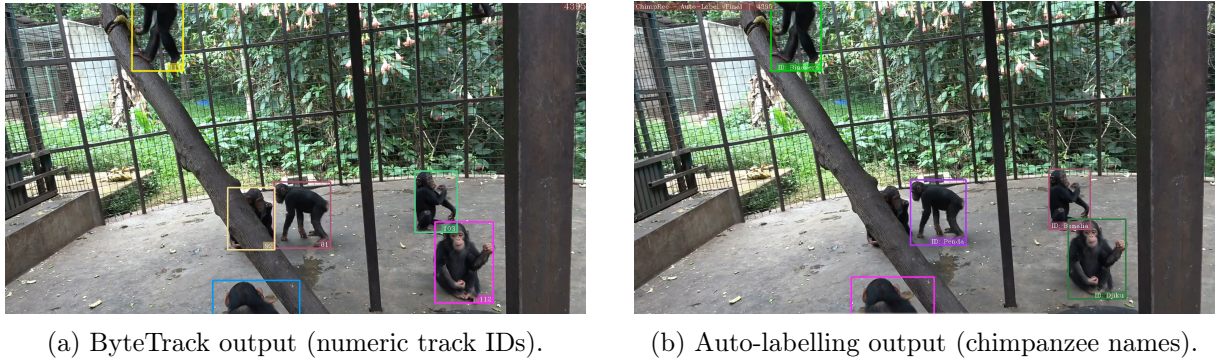


Figure 5.11: Matched frames from two supplementary YouTube fragments of Video K (11h33, evaluation video). Left: ByteTrack labels each detection with a numeric track ID that resets at every fragmentation. Right: the final auto-labelling pipeline replaces those IDs with the predicted chimpanzee names. *Click the images to watch each full clip (1 min 53 s).*

5.4.3 Residual Errors

The identity-aware metrics above evaluate whether the final labels are correct. A separate diagnostic is still needed for track continuity, because a trajectory can carry the right name while remaining fragmented across time. Table 5.9 therefore summarises the raw identity-switch and fragmentation counts for the three post-processing stages.

Table 5.9: Identity-continuity diagnostics for the three post-processing stages on the evaluation videos. Lower is better. Both post-processing stages reduce raw ID switches relative to ByteTrack, and the final auto-labelling pipeline additionally lowers fragmentations on both videos.

Stage	IDSW (J)	IDSW (K)	Frag. (J)	Frag. (K)
ByteTrack	611	430	1019	693
+ Dual-heuristic	394	209	1019	693
+ Auto-labelling (final)	350	253	911	651

Two patterns matter here. First, both post-processing stages cut the raw ID-switch count well below ByteTrack on both videos, with the dual-heuristic stage roughly halving it by relabelling fragmented tracklets under shared anonymous clusters, although it leaves fragmentations unchanged. Second, the final pipeline keeps switches low while also lowering fragmentations, so it improves both continuity axes at once. The residual switches that remain are tied to genuine tracker breakpoints rather than to inconsistent naming. Many fragments now carry the correct chimpanzee name on both sides of the break, which is why the qualitative output shows stable names through occlusions even when a tracklet is split. The remaining error is therefore mostly a track-continuity problem rather than a name-assignment problem.

5.5 Discussion

Old pipeline versus new pipeline. The single number that summarises this chapter is the HOTA of the end-to-end system on the evaluation videos before and after the contributions of this

thesis. On the evaluation pair (J, K) , the DeepSORT-inherited baseline obtains **HOTA** = 0.282 on Video J and 0.266 on Video K (Table 5.6), whereas the final auto-labelling pipeline of section 4.4.2 reaches **HOTA** = 0.628 on Video J and 0.521 on Video K (Table 5.7). Averaged across the two evaluation videos, this is a jump from 0.274 to 0.575, roughly a doubling of HOTA on footage that no component of the pipeline ever saw at training or tuning time. The rest of this section explains why this jump is concentrated in association rather than detection, why the absolute HOTA on Video K should not be over-interpreted, and what remains unsolved.

Figure 5.12 summarises the same progression in one curve. Tracker substitutions alone move the score only slightly. The first visible lift appears once the dual-heuristic post-processing baseline is introduced, and the large final step arrives only when the fine-tuned face model is integrated into the auto-labelling pipeline. In that sense, the figure is a compact narrative of the thesis progression. Detection was already serviceable, motion-only tracking reached its ceiling early, and the decisive gains came from identity-aware post-processing.

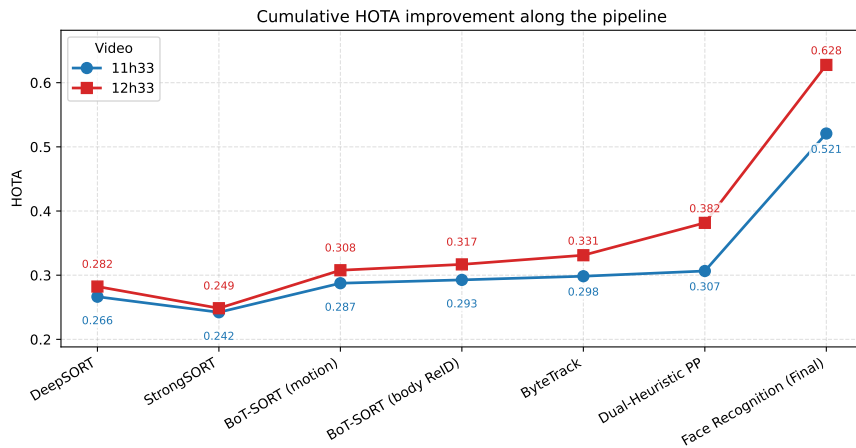


Figure 5.12: Cumulative HOTA improvement across the successive pipeline stages on evaluation Videos J (12h33) and K (11h33). The curve summarises the development arc of the thesis.

Detection versus association. Across the tracker and post-processing results, detection changes only moderately while association changes dramatically. The bottleneck of multi-object tracking on this footage is therefore not localisation but identity continuity. This conclusion holds regardless of the upstream tracker, which is the structural reason why an online motion tracker followed by a face-driven re-identification layer outperforms the monolithic alternatives tested here. The recognition table carries the same message from an identity-aware angle. The final gain is paid in correct *names*, not only in HOTA-friendly association statistics.

Effect of curated ground truth. Detection precision is lower on Video K than on Video J, and that gap is largely set by the annotation philosophy described earlier rather than by a simple detector failure. The absolute HOTA difference between the two evaluation videos should therefore not be read as a detector gap alone. It also reflects the fact that dense scenes contain more visually plausible boxes that are absent from the curated ground truth. The relative ordering between methods on each video is the part of the comparison that this curation policy

affects least, and it is consistent across J and K.

Remaining failure modes. The remaining errors fall into three regimes. First, long occlusions can split a true trajectory into fragments that receive inconsistent face evidence. Second, dense crossings can create short duplicate tracks that are difficult to resolve perfectly in every frame. Third, a small subset of visually similar or under-represented identities absorbs most of the residual name confusions. These regimes define the natural priority list for future work, longer-range temporal association, sharper duplicate handling, and targeted enrichment of the hardest identities in the face training corpus.

Chapter 6

Future Work

This chapter proposes concrete next steps grounded in the findings of section 5.5, organised by the layer of the pipeline they target.

6.1 Data and Annotation

Bootstrapped ground-truth generation. Now that the auto-labelling pipeline produces usable chimpanzee names on the evaluation videos, with IDAcc values of 0.700 on Video J and 0.781 on Video K, the obvious next step is to close the loop by using the pipeline output as a *first-pass annotation* that a human only needs to correct, rather than authoring from scratch. A useful prerequisite is to enrich the rendered output of Step 3 with the per-tracklet classifier margin of Equation (4.7), so that the annotator’s attention can be directed to the segments the pipeline itself flagged as low-confidence or as `unknown_n`. A quantitative measurement of the annotation-time reduction is also missing from the present thesis and would be a valuable contribution in its own right.

Targeted enrichment of hard identities. Table 5.3 and the discussion of section 5.2.3 identify identity-specific errors that cannot be explained by training-set size alone. Some identities, such as Kassongo and Maniema, combine limited clean Tier 1 support with difficult test examples, while the full-video analysis shows that other confusions are driven by deployment conditions. Long range, profile views, motion blur, and tight body contact can make a face crop much less informative than its curated test-set counterpart. These are two different failure modes and call for two different remedies. The first calls for additional curated Tier 1 crops for the weakest identities. The second calls for targeted Tier 3 hard-example mining, with validation safeguards, to include small or partially frontal faces that are currently under-represented. A controlled comparison of the two enrichment strategies, with the same training schedule, would clarify how much of the residual error is intrinsic to the face model and how much is an imaging artefact.

Outdoor and cross-site generalisation. The protocol of section 4.2.1 restricted tuning and evaluation to indoor footage on the grounds that the wire-mesh of the outdoor enclosure dominates the detector and tracker thresholds in a way that would bias the search. The trade-off is that the evaluation claim of Chapter 5 only covers indoor conditions. Re-running the same

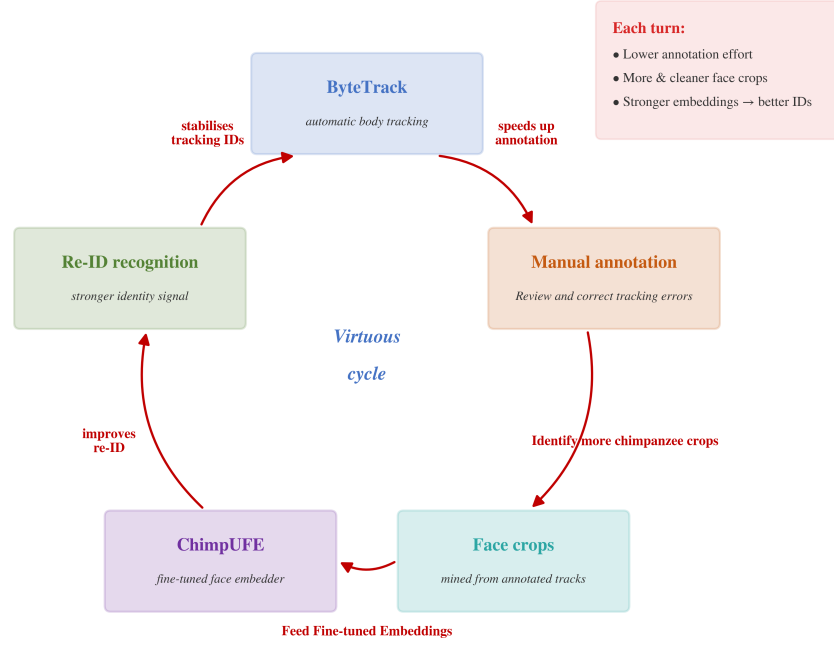


Figure 6.1: Bootstrapped improvement loop: auto-tracking supports faster manual correction, which increases face-crop yield and improves re-identification.

evaluation on a separate outdoor evaluation video, with the detector and tracker frozen at the indoor optima, would quantify how much of the gain transfers and which stage of the pipeline degrades first. A second axis is cross-site generalisation: the corpus of section 3.2 comes from a single facility, and the ArcFace head of section 4.3.2 is closed-set on its 20 identities. Deploying the same architecture on a new study group requires either retraining the head or operating it in open-set mode, which is discussed below.

6.2 Modelling

Longer-range temporal association. The residual ID switches summarised in Table 5.9 are consistent with difficult track breakpoints, especially long occlusions and dense crossings, but the present thesis does not include a dedicated occlusion-length analysis. Two directions can extend the association horizon. The first is to enlarge the ByteTrack buffer beyond its current value of 70 frames, with the obvious risk of admitting more false reattachments under camera motion. This would be most informative paired with the camera-motion compensation experiment described below. The second is to extend the embedding-based label-propagation stage 8 of the auto-labelling pipeline (see section 4.4.2) into a true gap-bridging operation that operates over multi-second interruptions rather than only between tracklets already separated by the tracker. A natural design is to maintain, per labelled cluster, a small bank of high-confidence face crops, and to evaluate candidate reattachments by their cosine similarity to the bank rather than to a single mean embedding.

Open-set face recognition. The current ArcFace head commits to one of the 20 known identities or abstains via `unknown_n`. The abstention rate is currently controlled by a single margin gate (Equations (4.7) and (4.8)), which conflates two qualitatively different reasons for refusing a name: a low-quality crop (no usable face visible) and a high-quality crop that nonetheless does not resemble any of the 20 prototypes (a new individual). Adapting the risk–coverage analysis of section 5.2.6 into an explicit out-of-distribution test on faces of chimpanzees absent from the training set would be the first step toward an operational open-set mode. The natural follow-up is to test whether a self-supervised backbone such as DINOv2 [20] or the original ChimpUFE checkpoint, used at inference without the ArcFace head, separates known and unknown identities more cleanly under cosine similarity.

Camera-motion compensation, revisited. In all online-tracker experiments reported in Chapter 4, BoT-SORT’s global motion compensation and StrongSORT’s ECC stabilisation were disabled to keep the comparison with motion-only ByteTrack clean. The decision was deliberate but leaves an open question: given that the footage is handheld with frequent pans and optical zoom, how much of the AssA gap could be closed by an explicit camera-motion model? A natural follow-up experiment is a controlled ablation along three axes: (i) the registration backend (sparse Lucas–Kanade as in BoT-SORT vs. dense ECC as in StrongSORT vs. a learned homography predictor), (ii) the warp regime (translation-only vs. affine vs. projective, to capture optical zoom in addition to pan), and (iii) the integration point (warping the Kalman state vs. warping the input frames before detection). Evaluating each variant on the evaluation pair (J, K) would also clarify whether the strict `match_thresh` = 0.9 selected in Table B.6 is a property of the footage or an artefact of operating without camera-motion compensation.

6.3 Pipeline-Level Refinements

Hyperparameter search over the auto-labelling gates. The gates collected in Table C.1 were set by hand from the diagnostic logs of Table C.2 and were never subjected to a Cartesian search of the size used for the trackers. Each gate is interpretable in isolation (per-tracklet margin, cluster-level margin, propagation cosine, merge cosine), and several of them are likely to be near-optimal, but the joint behaviour of the four classifier-related gates $(\tau_{\text{cls}}, \delta_{\text{cls}}, \tau_{\text{cls}}^c, \delta_{\text{cls}}^c)$ on the tuning videos has not been mapped. A two-phase search analogous to the tracker protocol, with the evaluation pair (J, K) kept strictly outside the search loop, would be a useful confirmation that the production gates are not local optima of the version-by-version exploration.

Sharper per-frame deduplication. The per-frame deduplication of stage 9 (see section 4.4.2) resolves same-name conflicts by keeping the highest-confidence box on each frame and dropping the others. The diagnostics of section 5.4.3 indicate that on a small number of frames the second-best box happens to win, producing the dense crossings that dominate the residual ID-switch count summarised in Table 5.9. Two refinements are worth testing: (i) aggregating the per-frame decision over a short window so that an isolated drop in classifier confidence cannot swap the assignment, and (ii) using the IoU history with the prior frame’s winner as a tie-breaker,

which reuses the temporal-conflict guard already implemented in the dual-heuristic pipeline of section 4.4.1.

6.4 Behavioural Downstream Tasks

User-facing annotation interface. Another practical extension is to wrap the current pipeline in a lightweight web application designed for end users who do not have a computer-science background. At present, running the pipeline, inspecting intermediate outputs, and correcting annotations still requires handling scripts and configuration files that are not well suited to routine use by biologists. A web interface could expose the core workflow through a simpler sequence of actions, selecting a video, launching the automatic pipeline, visualising the resulting tracks and names, reviewing low-confidence segments, and exporting corrected annotations. Such a layer would not change the underlying algorithms, but it would make the system substantially easier to adopt in practice and would help turn the current research code into a more usable annotation tool for behavioural studies.

The original motivation of this thesis, stated in Chapter 1, is to support a doctoral research project on the social development of juvenile chimpanzees. The output of the auto-labelling pipeline (per-frame body boxes tagged with chimpanzee names) is the natural input of every behavioural analysis built on co-occurrence, proximity, or social network structure. A first concrete deliverable is a co-occurrence and dyadic-proximity report computed directly from the auto-labelled tracks on the eleven annotated videos, validated against the manually corrected ground truth. A second, more demanding deliverable is the detection of social play events themselves, for which the per-tracklet face evidence collected by the pipeline could serve as an auxiliary signal alongside the body trajectories. Both deliverables sit at the interface of the present thesis and the biologist’s research programme, and both would benefit from the auto-labelling output being ingested by an analyst rather than by another vision module.

Chapter 7

Conclusion

7.1 Summary of the Thesis

This thesis revisited the chimpanzee identification problem inherited from the **ChimpRec** project [1] and transformed it into a complete tracking, recognition, and auto-labelling workflow. The objective was not only to improve a benchmark score, but to move toward a practical annotation system that can support biologists working with long, identity-rich videos.

The central challenge is that chimpanzees are visually similar, frequently occlude one another, and are filmed with a handheld camera. Under these conditions, this thesis shows that the main bottleneck is not body localisation alone, but long-term identity association. The final pipeline therefore moves away from a purely tracker-centred formulation and adopts a two-stage design. ByteTrack first follows chimpanzee bodies frame by frame, and a later offline post-processing stage assigns identities by accumulating face evidence over the resulting tracklets.

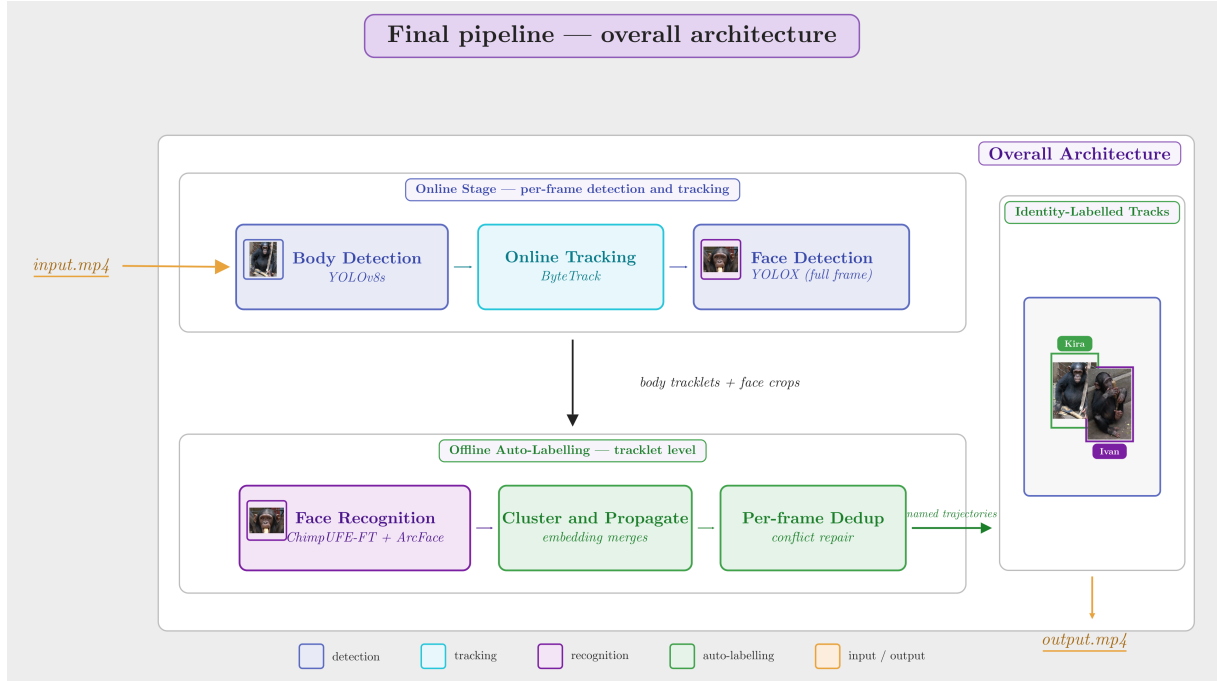


Figure 7.1: Final end-to-end architecture delivered in this thesis.

7.2 Main Contributions

The first contribution was infrastructural. A three-step annotation workflow, a curated eleven-video ground-truth corpus, and a HOTA-compatible evaluation layer were built so that tracking variants could be compared under a single reproducible protocol. This infrastructure also made it possible to mine additional body and face crops from the annotated videos. The resulting three-tier face corpus combined curated identity crops, student annotations, and automatically mined video crops, giving the face-recognition stage enough diversity to move beyond the limitations of the previous pipeline.

The second contribution was the fine-tuning of ChimpUFE for the study group. Starting from the publicly released self-supervised ChimpUFE backbone, the thesis added an ArcFace identity head and trained it with tier-aware sampling and tier-weighted loss. This adaptation turned the pretrained face embedder into a population-specific recognition model able to provide reliable identity evidence for downstream post-processing.

The third contribution was the end-to-end auto-labelling pipeline. The tracker comparison established that body-based Re-ID, even after domain-specific fine-tuning, does not provide enough discriminative information to solve association at frame level. The final architecture therefore decouples localisation from identification. ByteTrack handles body localisation and short-term motion, while a face-driven post-processing layer accumulates evidence over tracklets, assigns chimpanzee names, merges compatible fragments, propagates labels, repairs duplicate conflicts, and interpolates short gaps.

7.3 Main Findings

The experiments lead to three main findings.

First, online tracking alone does not solve the chimpanzee identification problem. ByteTrack, BoT-SORT, StrongSORT, and DeepSORT can localise bodies with reasonable accuracy, but all leave a substantial association gap. The HOTA decomposition shows that detection performance is consistently stronger than association performance. This confirms that the dominant difficulty is not simply finding chimpanzees in each frame, but maintaining consistent identities through occlusions, crossings, and re-detections.

Second, body appearance is not sufficiently informative at frame level. Fine-tuning OSNet on chimpanzee body crops improves domain adaptation, but the resulting body Re-ID embeddings remain too weak to drive reliable online association. This supports the central design choice of the thesis: the system should not rely on body appearance as the main identity cue when the discriminative information is mostly located on the face.

Third, face recognition becomes useful once the model is fine-tuned on the study group and its predictions are combined over several frames. On the balanced face test set, the fine-tuned ChimpUFE–ArcFace model reaches 85.4% Rank-1 accuracy, compared with 27.6% for the pretrained ChimpUFE backbone and 7.2% for the inherited FaceNet-FC baseline. This improvement turns face information from a weak auxiliary cue into a usable identity signal. Integrated into the final auto-labelling pipeline, this identity signal produces the main end-to-end improvement. On evaluation videos J and K, the final pipeline reaches HOTA scores of 0.628 and 0.521, compared with 0.282 and 0.266 for the inherited DeepSORT baseline. It also reaches IDAcc scores of 0.700 and 0.781, respectively. The analysis shows that these gains come primarily from improved association and name consistency, rather than from detecting many additional animals.

Together, these findings support the main scientific conclusion of the thesis: robust chimpanzee video annotation should not be treated as a purely online tracking problem. A more effective formulation is to separate body localisation from identity assignment, and to let sparse but discriminative face evidence be accumulated over time.

7.4 Limitations and Future Directions

The resulting system should not be interpreted as a fully automatic solution. The evaluation remains limited to indoor footage from a single facility, and the face recogniser is closed-set over the twenty known individuals of the study group. The auto-labelling gates were selected manually, and the pipeline still fails in difficult cases such as long occlusions, dense crossings, poor face visibility, and extreme camera motion.

The curated nature of the ground truth also affects the interpretation of the metrics. The annotation files were designed to be useful for biological analysis, not to preserve every barely visible animal in every frame. As a result, absolute detection precision must be interpreted carefully, especially in dense scenes where the automatic pipeline may keep visually correct but uncurated

boxes. The reported scores are therefore best understood as comparative measurements under a fixed annotation policy.

These limitations define the next steps rather than diminishing the contribution of the work. Future improvements should focus on bootstrapped annotation, targeted enrichment of difficult identities, open-set recognition, camera-motion compensation, systematic tuning of auto-labelling gates, and a user-facing review interface. Each of these directions follows directly from the residual errors identified in the analysis chapter.

7.5 Final Perspective

The main outcome of this thesis is a transition from manual correction after tracking to semi-automatic, identity-aware annotation. The delivered pipeline can already produce first-pass identity labels that a human annotator can inspect and correct. Each corrected video can then feed the next round of training data, creating a feedback loop between annotation, model improvement, and biological use.

This feedback loop is the most important practical impact of the work. It creates a path toward larger annotated corpora, stronger recognition models, and downstream behavioural analyses based on co-occurrence, proximity, and social interaction. In this sense, the thesis does not deliver a finished deployment system, but a reproducible operational foundation that the biological research programme can now use, evaluate, and extend.

It turns `ChimpRec` from a tracker followed by manual correction into the first iteration of a scalable annotation loop for individual-based chimpanzee video analysis.

Appendix A

Keypoint-Based Approach

A.1 Motivation

Before converging on the bounding-box pipeline used in the main thesis, we briefly investigated whether pose estimation could provide a richer representation of chimpanzee motion. Instead of representing each animal by an axis-aligned box, a pose-based tracker predicts anatomical keypoints and links the resulting skeletons over time. In principle, such a representation could encode posture, limb configuration, and body orientation more directly than a crop-level Re-ID model.

We tested this idea with SLEAP [34], a multi-animal pose-estimation framework that provides an annotation interface, top-down and bottom-up neural pipelines, and motion-based tracking utilities. Our experiment used the top-down SLEAP pipeline, where a centroid network first localises each animal and a centered-instance network then predicts the keypoints inside each crop. Identity heads were not used, because they require identity-labelled skeleton data at training time.

A.2 Data and Training

A.2.1 Skeleton Design and Annotation

We defined a 9-keypoint skeleton focused on the upper body and the most visible limb extremities, *head*, *neck*, *torso*, left and right *shoulder*, left and right *wrist*, and left and right *ankle*. The torso was used as the anchor part for the top-down pipeline because it remains visible in more poses than the head or limbs.

Two annotation passes were performed in the SLEAP GUI. The first labelled 39 frames sampled from the ground-truth video corpus, covering isolated poses, climbing, partial occlusion, and multi-individual interactions. A second pass extended the set to 98 labelled frames. This revealed the main practical limitation of the approach which is that a skeleton frame was much slower and more ambiguous to annotate than a bounding box, especially when limbs were occluded or intertwined during play.

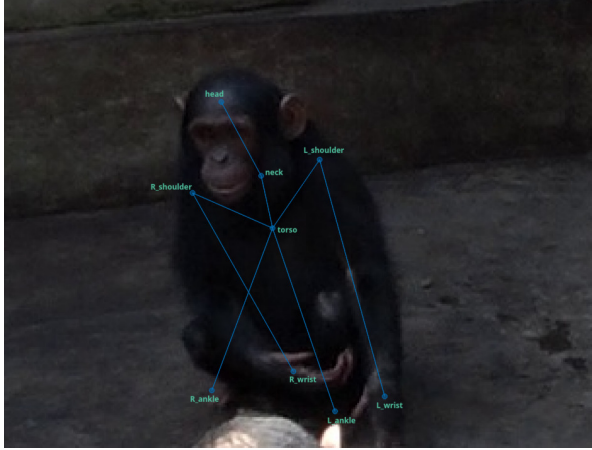


Figure A.1: Two manually annotated SLEAP frames. **Left:** an isolated juvenile, close to the best-case setting for keypoint placement. **Right:** two interacting individuals, where overlapping limbs and partial occlusion already require annotator interpretation.

A.2.2 Training and Inference

Both SLEAP networks used the standard top-down U-Net configuration recommended for small multi-animal corpora. The centroid model was trained on half-resolution inputs, while the centered-instance model operated at full resolution. The main runs used a 90/10 split of the 39-frame corpus. The centroid network stopped after 24 epochs and the centered-instance network after 28 epochs. A second pair of runs on the 98-frame corpus stopped after only 11 epochs with worse losses, so the larger annotation pass did not translate into better inference quality.

Table A.1: Summary of the SLEAP side experiment. The 98-frame run was attempted but did not improve over the smaller converged run.

Run	Labelled frames	Epochs	Outcome
Centroid network	39	24	Converged, limited generalisation
Centered-instance network	39	28	Converged, unstable in crowded frames
Extended corpus	98	11	Early stop, worse losses

Figure A.2 shows the typical qualitative outcome. Static isolated individuals could be localised acceptably, but skeletons fragmented, jumped between animals, or lost limbs as soon as individuals touched, crossed, or moved under camera motion.



Figure A.2: Representative SLEAP inference frame. *Click the image to watch the full qualitative inference video on YouTube.*

A.3 Why the Pipeline Was Abandoned

The SLEAP experiment was abandoned for three reasons.

- **Annotation cost.** Skeleton annotation was roughly an order of magnitude slower than bounding-box annotation and required anatomical decisions for occluded landmarks. Scaling to the few hundred frames needed for robust pose estimation was not realistic within the thesis scope.
- **Non-rigid articulation.** Juvenile chimpanzees frequently climb, hang, curl up, or interact in poses where keypoint placement is ambiguous even for a human annotator, which produces noisy targets and unstable predictions.
- **Occlusion and close contact.** Social play and the enclosure geometry hide limbs and create overlapping bodies. The resulting partial skeletons are too fragile to support reliable tracking or identity association.

Given these limitations, and given the stronger robustness of YOLO-based body detection on the same footage, all main tracking and identity experiments in the thesis use bounding-box detections rather than pose skeletons.

Appendix B

Tracker Hyperparameter Search Details

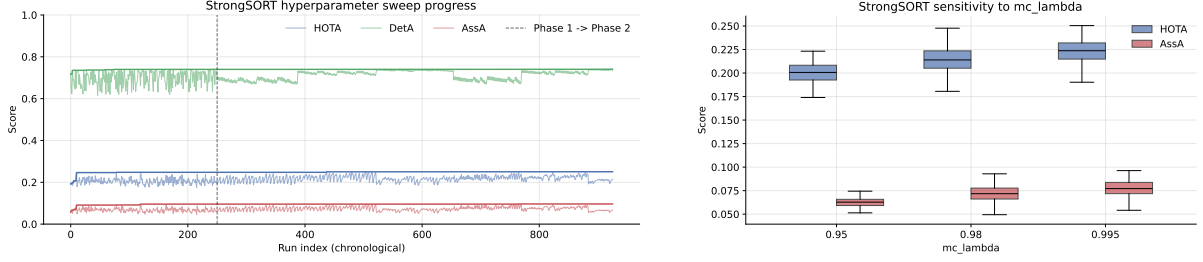
This appendix collects the detailed per-tracker search spaces, the two-phase tuning budgets, and the bookkeeping figures (number of configurations evaluated, embedding pre-computation costs) that complete the description of the tracking experiments of section 4.2. The cached-replay protocol, the tuning videos, the shared optimisation criterion, and the two-phase search strategy are introduced in section 4.2.1. The notation and parameter names follow the main text. The interpretation of these diagnostics is kept in sections 5.3.2 and 5.3.3 so that the appendix remains a technical record rather than a second results chapter.

B.1 StrongSORT

Seven hyperparameters were swept (Table B.1), defining a Cartesian product of $4 \times 3 \times 3 \times 4 \times 3 \times 3 \times 3 = 3\,888$ configurations. The two-phase strategy comprised:

1. **Phase 1 - Random exploration.** 250 configurations sampled uniformly from the full grid.
2. **Phase 2 - Local refinement.** For each of the top-5 Phase 1 configurations, a ± 1 -step neighbourhood was generated, yielding 1 116 unique configurations.

Due to an under-estimated wall-clock limit on the submitted SLURM job, the run was killed after completing all 250 Phase 1 configurations and 676 of the 1 116 Phase 2 configurations, for a total of **926 evaluated configurations**. Both tuning videos (A: 07h56 and E: 14h29) were scored for every configuration. The resulting scores and their interpretation are reported in sections 5.3.2 and 5.3.3.



(a) HOTA across the five independent SLURM tuning jobs. Each point is the mean HOTA over the two tuning videos. Bars show the min/max range within the job.

(b) Marginal sensitivity to `mc_lambda`, the weight fusing motion and appearance costs. Higher values favour pure motion.

Figure B.1: StrongSORT hyperparameter search diagnostics. Left: per-job HOTA spread. Right: marginal effect of `mc_lambda` on HOTA, DetA and AssA across all evaluated configurations.

Table B.1: StrongSORT hyperparameter search space. The full Cartesian product contains 3 888 configurations. The two-phase search evaluated 926 of them.

Parameter	Values	Description
<code>detection_conf</code>	0.3, 0.4, 0.5, 0.6	YOLO minimum confidence filter
<code>ema_alpha</code>	0.85, 0.90, 0.95	EMA decay for track appearance state
<code>max_age</code>	50, 100, 200	Frames before track removal
<code>max_cos_dist</code>	0.15, 0.20, 0.30, 0.40	Cosine-distance gate on appearance
<code>max_iou_dist</code>	0.5, 0.7, 0.9	IoU-distance gate in cascade fallback
<code>mc_lambda</code>	0.95, 0.98, 0.995	Motion/appearance fusion weight
<code>n_init</code>	2, 3, 5	Confirmations before a track is published

B.2 BoT-SORT

The five motion-related parameters were fixed to the optimum found during the ByteTrack tuning (Table B.6). The appearance-focused search covers the five free parameters listed in table B.2, defining a Cartesian product of $3 \times 4 \times 6 \times 5 \times 3 = 1\,080$ combinations. A motion-only baseline (BoT-SORT with Re-ID disabled) was evaluated by exhaustively sweeping the three shared parameters `match_thresh`, `track_buffer` and `detection_conf`, yielding $3 \times 4 \times 3 = 36$ additional combinations.

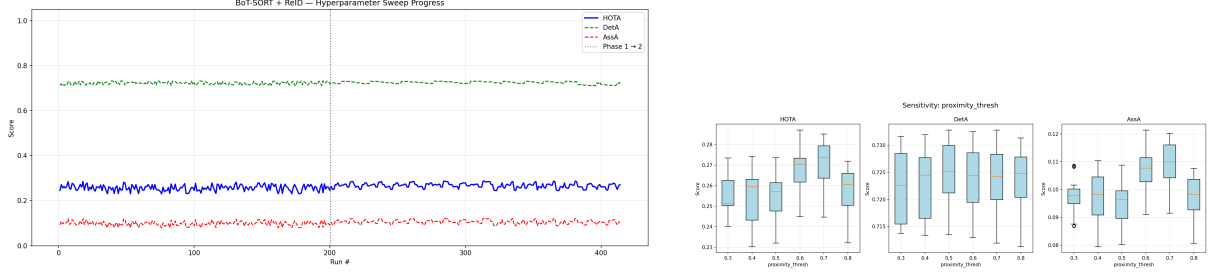
Table B.2: BoT-SORT hyperparameter search space. Motion parameters are fixed at the optimal ByteTrack configuration. Only appearance-related and shared parameters are tuned.

Parameter	Values	Description
<code>match_thresh</code>	0.7, 0.8, 0.9	IoU cost matching threshold
<code>track_buffer</code>	50, 70, 100, 150	Frames before track removal
<code>proximity_thresh</code>	0.3, 0.4, 0.5, 0.6, 0.7, 0.8	Spatial gate for appearance
<code>appearance_thresh</code>	0.15, 0.25, 0.4, 0.6, 0.8	Minimum appearance similarity
<code>detection_conf</code>	0.3, 0.4, 0.5	YOLO minimum confidence filter

The two-phase strategy comprised: (i) Phase 1, 200 configurations randomly sampled from the 1 080 candidate combinations, plus the 36 motion-only baselines evaluated exhaustively. (ii) Phase 2, a ± 1 -step neighbourhood around the top-5 Phase 1 configurations, generating

214 new combinations. In total, **414 configurations** were evaluated.

The pre-computation phase required extracting Re-ID embeddings for all cached detections: 126 164 embeddings for video A and 67 247 for video E. This one-time ReID extraction added approximately 45 minutes of GPU time per video (batch size 128 on an NVIDIA RTX 6000 Ada), but all subsequent tracker replays remained lightweight (< 30 seconds each).



(a) Sweep progress for BoT-SORT + Re-ID. The Phase 1→2 boundary is shown by the dotted vertical line. HOTA is nearly flat across all configurations, indicating that the appearance branch does not improve tracking over the motion-only baseline.

(b) Marginal sensitivity to the proximity threshold θ_{IoU} . The boxplots span all configurations evaluated in Phase 1.

Figure B.2: BoT-SORT search diagnostics. Left: per-run HOTA/DetA/AssA. The flat profile confirms that appearance embeddings do not provide a consistent signal on our footage. Right: sensitivity to the spatial gate θ_{IoU} .

B.3 ByteTrack

Seven parameters were swept (Table B.3), defining a total search space of $4 \times 3 \times 4 \times 5 \times 3 \times 2 \times 4 = 5\,760$ configurations.

Table B.3: ByteTrack hyperparameter search space. The Cartesian product contains 5 760 configurations.

Parameter	Values	Description
<code>track_high_thresh</code>	0.3, 0.4, 0.5, 0.6	First-stage association threshold
<code>track_low_thresh</code>	0.05, 0.1, 0.2	Second-stage threshold
<code>new_track_thresh</code>	0.4, 0.5, 0.6, 0.7	Minimum confidence for new tracks
<code>track_buffer</code>	30, 50, 70, 100, 150	Frames before track removal
<code>match_thresh</code>	0.7, 0.8, 0.9	IoU cost matching threshold
<code>fuse_score</code>	True, False	Fuse detection confidence into cost
<code>detection_conf</code>	0.3, 0.4, 0.5, 0.6	YOLO minimum confidence filter

The two-phase strategy comprised: (i) Phase 1, 372 configurations randomly sampled from the full space, each evaluated on both tuning videos with HOTA, DetA and AssA logged. (ii) Phase 2, a ± 1 -step neighbourhood around the top-5 Phase 1 configurations. HOTA matching used the Hungarian algorithm across the 19 IoU thresholds $\alpha \in \{0.05, 0.10, \dots, 0.95\}$ used by the evaluation pipeline.

The detection pre-computation step was run at $\text{conf} \geq 0.1$ (the minimum in the search space), producing 126 164 detections for video A and 67 247 for video E. Higher `detection_conf` values

during tracker replay simply filter this cached set, ensuring strictly comparable detection inputs across all configurations.

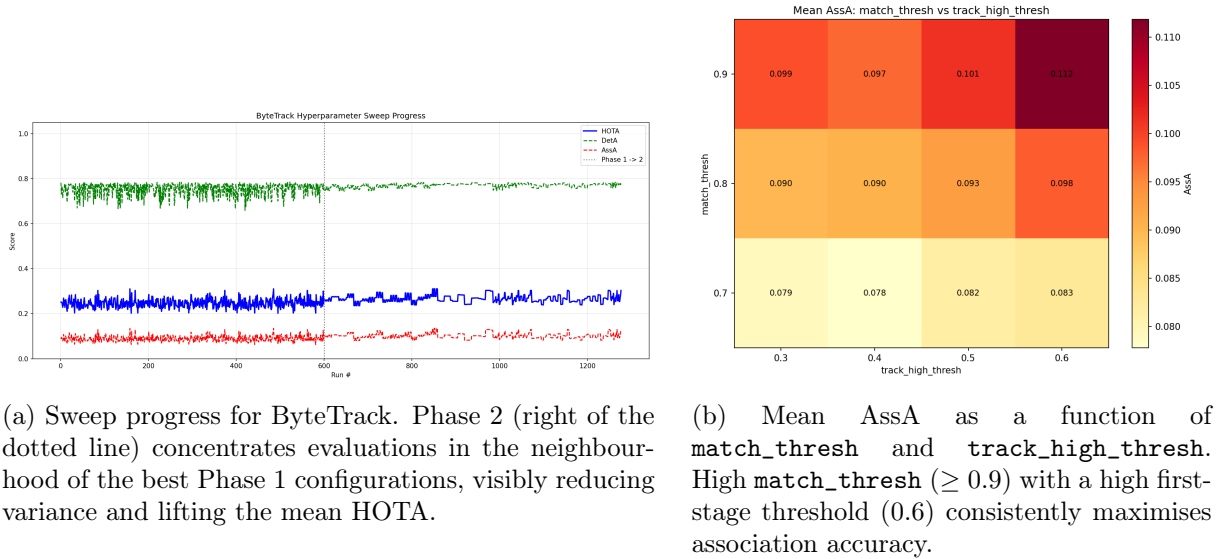


Figure B.3: ByteTrack search diagnostics. Left: per-run HOTA/DetA/AssA across Phase 1 and Phase 2. Right: two-parameter interaction heatmap showing mean AssA as a function of the two most influential parameters.

B.4 Selected Configurations

For each tracker, the configuration retained at the end of the two-phase search is summarised below. The mean HOTA is the value obtained on the tuning videos (A, E). It is not directly comparable to the evaluation HOTA of table 5.6, which is computed on Videos J and K .

Table B.4: StrongSORT best configuration. Search: 926 configurations evaluated out of 3 888 across the two-phase strategy.

detection_conf	0.5
ema_alpha	0.95
max_age	200
max_cos_dist	0.30
max_iou_dist	0.70
mc_lambda	0.995
n_init	5
Mean HOTA (tuning set A, E)	0.250

Table B.5: BoT-SORT (with chimp-fine-tuned OSNet) best configuration.

detection_conf	0.5
match_thresh	0.9
track_buffer	70
proximity_thresh	0.6
appearance_thresh	0.15
Mean HOTA (tuning set A, E)	0.287

Table B.6: ByteTrack best configuration. The five motion parameters identified here are also reused, frozen, by BoT-SORT (Section 4.2.5) since BoT-SORT inherits ByteTrack’s association logic.

detection_conf	0.5
track_high_thresh	0.6
track_low_thresh	0.1
new_track_thresh	0.7
track_buffer	70
match_thresh	0.9
fuse_score	true
Mean HOTA (tuning set A, E)	0.311

Appendix C

Auto-Labelling Pipeline Hyperparameters

C.1 Post Processing Selected Configurations

Stage	Symbol	Value	Role
2	σ_{∇}^2 floor	25	Laplacian sharpness floor for face crops
2	size floor	0.005	Min $A_{\text{face}}/A_{\text{body}}$
2	face pool	60	Frames sampled per tracklet
2	face top- k	12	Crops kept per tracklet
3	τ_{cls}	0.40	Per-tracklet top-1 gate
3	δ_{cls}	0.05	Per-tracklet top-1/top-2 margin
4	τ_{ovl}	5	Overlap tolerance (frames)
4	τ_{IoU}	0.50	Phantom-duplicate IoU
5	τ_{merge}	0.55	Greedy single-link cosine gate
7	τ_{cls}^c	0.22	Cluster-aggregated top-1 gate
7	δ_{cls}^c	0.04	Cluster-aggregated margin
7	min cluster crops	6	Crops required to re-classify a cluster
8	τ_{prop}	0.50	Label-propagation cosine gate
10	gap	30	Max gap (frames) for interpolation

Table C.1: Hyperparameters of the auto-labelling pipeline.

The Stage 2 crop-quality thresholds in Table C.1 are the production auto-labelling thresholds. They are deliberately less restrictive than the Tier 3 training-data filter of section 3.5.5, because auto-labelling ranks and aggregates several crops per tracklet instead of turning every retained crop into a training sample.

Version	Ingredient added	Cumulative pipeline
v2	YOLOX face-in-frame cascade, composite crop score	1–5, 11
v3	Spatial tracklet stitching with weak confidence	+early stitching
v4	Sharpness floor + IoU-aware duplicate guards	1–5, 11
v5	Embedding-based label propagation	1–5, 8, 11
v6	Cluster-aggregated re-classification	1–8, 11
v7	Whole-cluster demotion on same-name conflicts	1–9 (whole-cluster)
v8	IoU-aware exclusion of phantom hand-offs from the conflict count	idem, refined
v9	Fallback reassignment for the losing cluster	idem, softened
v10	Globally tightened gates ($\tau_{\text{merge}}, \tau_{\text{cls}}^c, \tau_{\text{prop}}$)	idem, conservative
v11	Per-frame deduplication in place of whole-cluster demotion	1–11 (production)
final	Self-contained refactor (no inter-version imports)	1–11 (production)

Table C.2: Incremental development of the auto-labelling pipeline. “Cumulative pipeline” lists the stages of Figure 4.4 that are active in each version. v11 is the configuration shipped as the final version.

Appendix D

Tier Statistics and Tier-Aware Contribution Computation

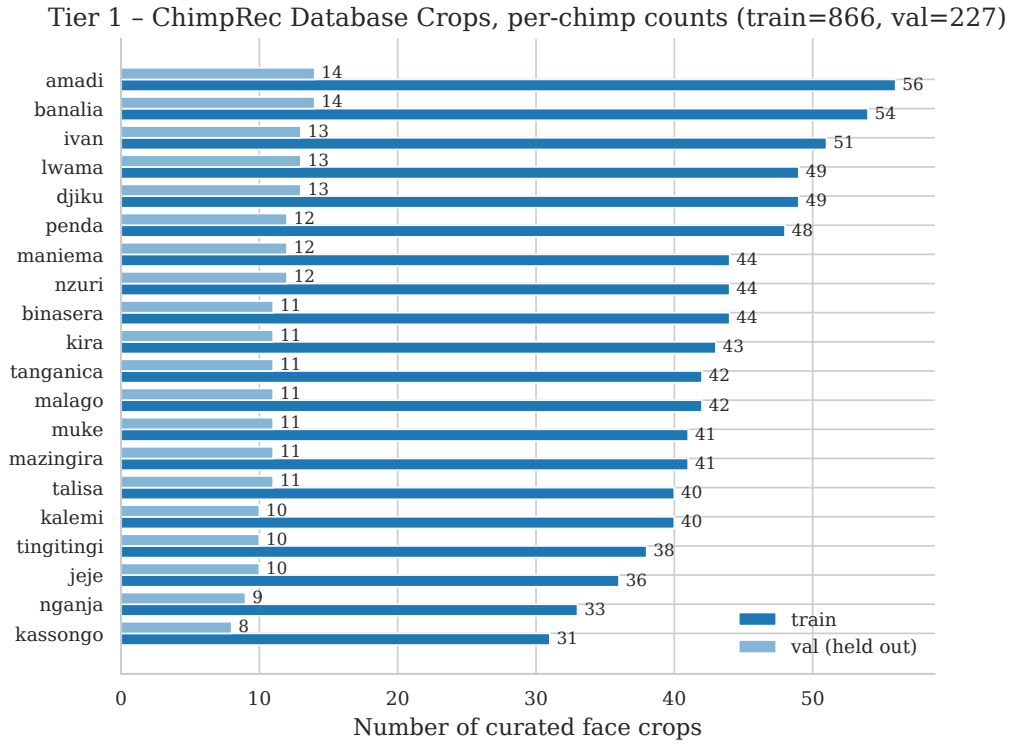


Figure D.1: Per-individual image counts in the Tier 1 curated gallery, split by training and validation partition. The validation split is fixed for all fine-tuning experiments.

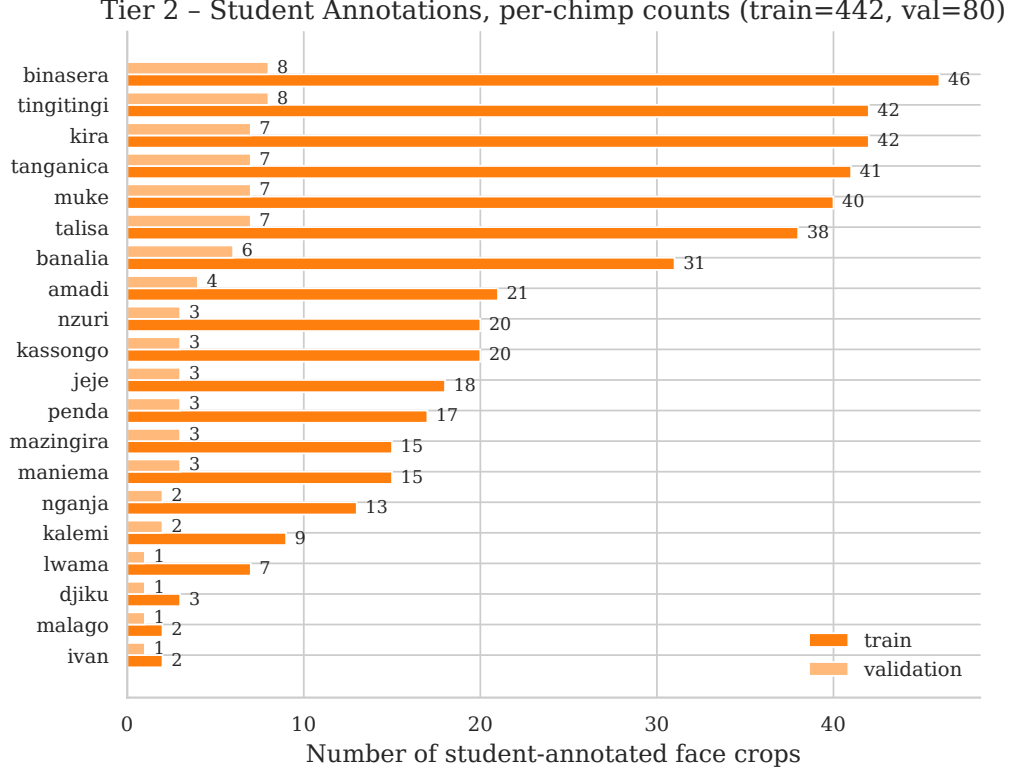


Figure D.2: Per-individual image counts in the Tier 2 student annotation database, showing the deterministic 85 / 15 train/validation split.

D.1 Expected Tier Contribution Computation

This appendix details how we compute the expected tier contributions reported as ρ_t in Table 4.1. These quantities are intended as an interpretable summary of the combined effect of tier-aware sampling and tier-weighted loss. They should be interpreted as *expected loss-weighted sampling contributions*, not as exact realised gradient shares, since the actual gradient also depends on prediction error, image difficulty, and the ArcFace margin.

D.1.1 Sampling probabilities after resampling cap

Sampling is performed with replacement using a `WeightedRandomSampler` over a virtual epoch of

$$N_{\text{epoch}} = |\mathcal{D}_{\text{train}}| \quad (\text{D.1})$$

draws. Each image i has an initial unnormalised weight w_i defined in Equation (4.2). To limit excessive oversampling, we apply a global cap (Section D.1.2), which produces final clamped weights $\tilde{w}_i$. After clamping, the probability of drawing image i is

$$p_i = \frac{\tilde{w}_i}{\sum_j \tilde{w}_j}. \quad (\text{D.2})$$

D.1.2 Global resampling cap

Before clamping, the expected number of draws of image i in one virtual epoch is

$$E_i = N_{\text{epoch}} \frac{w_i}{\sum_j w_j}. \quad (\text{D.3})$$

If $E_i > R$, the corresponding weight is reduced so that the expected reuse does not exceed the cap. The clipping is applied iteratively because reducing the largest weights changes the normalisation constant. The sampler therefore uses $\tilde{w}_i$ rather than w_i .

D.1.3 Expected loss-weighted contributions

Let $\mathcal{D}_{c,t}$ be the set of training images of chimpanzee identity c belonging to tier t , and let λ_t be the tier loss weight used in Equation (4.3). We define the expected loss-weighted sampling contribution for (c, t) as

$$I_{c,t} = \lambda_t \sum_{i \in \mathcal{D}_{c,t}} N_{\text{epoch}} \frac{\tilde{w}_i}{\sum_j \tilde{w}_j}. \quad (\text{D.4})$$

This corresponds to the expected number of sampled views from tier t for identity c in one epoch, scaled by the tier loss weight.

For each identity c , the relative contribution of tier t is

$$\rho_{c,t} = \frac{I_{c,t}}{\sum_{t'=1}^3 I_{c,t'}}, \quad (\text{D.5})$$

so that $\rho_{c,1} + \rho_{c,2} + \rho_{c,3} = 1$.

We also report a global tier share by aggregating over all images of each tier. Let $\mathcal{D}_t$ be the set of training images in tier t . The global expected tier contribution is

$$\rho_t = \frac{\lambda_t \sum_{i \in \mathcal{D}_t} \tilde{w}_i}{\sum_{t'=1}^3 \lambda_{t'} \sum_{i \in \mathcal{D}_{t'}} \tilde{w}_i}. \quad (\text{D.6})$$

The values of ρ_t are computed using the same clamped weights $\tilde{w}_i$ as the training sampler and provide a compact summary of the intended balance between curated manual tiers and the larger but noisier mined tier.



Figure D.3: Expected loss-weighted training contribution of each tier for each chimpanzee, after class balancing, tier-aware sampling, and the resampling cap. Remaining differences across individuals reflect variations in class composition and tier distribution.

Appendix E

Additional Face-Model Diagnostics

E.1 Confusion Matrix Face Recognition Models

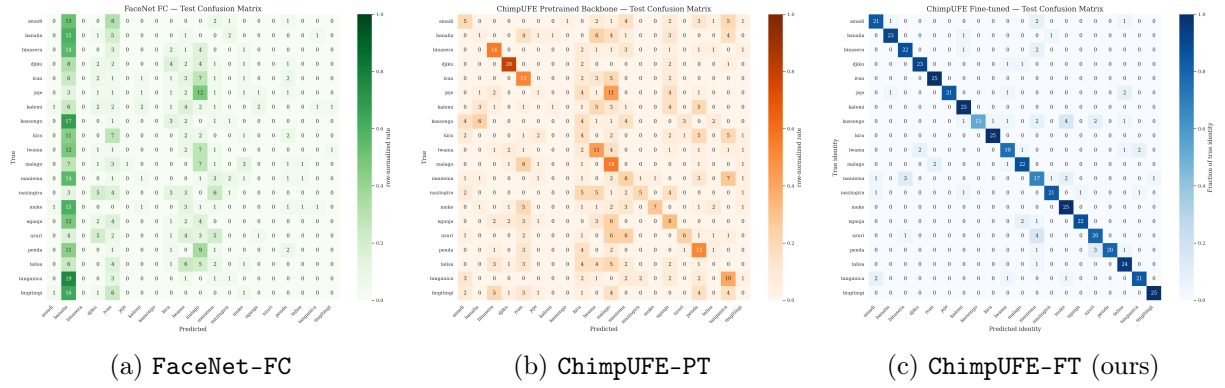


Figure E.1: Confusion matrices on the test set (Section 3.5.6) for the three evaluated face models. The **ChimpUFE-FT** model has more correct per-identity predictions, the **ChimpUFE-PT** shows more off-diagonal mass. The **FaceNet-FC** baseline shows a strong class bias, with many samples mapped to a single predicted identity (**Banalia**).

E.2 Training-Set Size Diagnostic

To complement the Tier 3 ablation of section 5.2.4, table E.1 reports Spearman correlations between training support and per-identity recall/precision. Spearman correlation measures whether two variables tend to increase or decrease together in a monotonic way, without assuming a linear relationship. It is therefore appropriate here because the number of training images varies strongly across identities, especially for Tier 3. This analysis separates the two components of F1. Tier 1 and Tier 2 training sizes are not clearly associated with recall. Tier 3 and total training size show moderate positive associations with recall, but the same data features are almost unrelated to precision. This pattern suggests that additional mined crops may help the model recover more true examples of an identity, while false-positive behaviour is not well explained by the number of training images alone. Since this analysis is based on only 20 identities, the correlations should be interpreted as diagnostic evidence rather than as a definitive statistical model.

Table E.1: Spearman correlation between training data availability and per-identity precision/recall. Spearman correlation measures monotonic association and does not assume a linear relationship, which is useful here because the number of training images varies strongly across identities.

Data feature	Recall		Precision	
	Spearman ρ	p -value	Spearman ρ	p -value
Tier 1 train images	-0.066	0.782	-0.294	0.209
Tier 2 train images	0.168	0.479	0.024	0.919
Tier 3 train images	0.447	0.048	0.047	0.845
Total train images	0.457	0.043	0.044	0.855

E.3 Model Agreement

A comparison between the ChimpUFE pretrained backbone and ChimpUFE fine-tuned model (ArcFace) is given in table E.2. The main observation is the strong asymmetry between the two models: when only one model is correct, it is almost always the ChimpUFE fine-tuned model (ArcFace). The ChimpUFE pretrained backbone is correct on only two images missed by the ChimpUFE fine-tuned model. The remaining cases where both models fail correspond mostly to the hardest part of the test set and overlap with the identities discussed in the table 5.3 of the results chapter.

Table E.2: Agreement between ChimpUFE-FT and ChimpUFE-PT on the 500 test images

Outcome	Count	Share
Both correct	137	27.4%
ChimpUFE-FT only correct	290	58.0%
ChimpUFE-PT only correct	2	0.4%
Both wrong, same prediction	35	7.0%
Both wrong, different prediction	36	7.2%

This agreement analysis confirms that the improvement is not limited to a small subset of images. ChimpUFE-FT corrects 290 images that ChimpUFE-PT misclassifies, while the reverse happens for only two images.

Bibliography

- [1] T. Cousin and J. Demeure, “Chimpanzee multi-object tracking: Detection, tracking, and identity recognition,” master’s thesis, Université catholique de Louvain, 2025.
- [2] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, “HOTA: A higher order metric for evaluating multi-object tracking,” *International Journal of Computer Vision*, 2021.
- [3] V. Iashin, H. Lee, D. Schofield, and A. Zisserman, “Self-supervised learning on camera trap footage yields a strong universal face embedder,” in *2025 IEEE International Conference on Image Processing Challenges and Workshops (ICIPCW)*, IEEE, 2025.
- [4] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *CoRR*, vol. abs/1506.02640, 2015.
- [5] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: exceeding YOLO series in 2021,” *CoRR*, vol. abs/2107.08430, 2021.
- [6] M. Yaseen, “What is yolov8: An in-depth exploration of the internal features of the next-generation object detector,” 2024.
- [7] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [8] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, pp. 83–97, 1955.
- [9] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” *CoRR*, vol. abs/1703.07402, 2017.
- [10] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE International Conference on Image Processing (ICIP)*, p. 3464–3468, IEEE, 2016.
- [11] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, “StrongSORT: Make DeepSORT great again,” *IEEE Transactions on Multimedia*, 2023.
- [12] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, “Bag of tricks and a strong baseline for deep person re-identification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

- [13] G. D. Evangelidis and E. Z. Psarakis, “Parametric image alignment using enhanced correlation coefficient maximization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858–1865, 2008.
- [14] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, “Bytetrack: Multi-object tracking by associating every detection box,” *arXiv preprint arXiv:2110.06864*, 2022.
- [15] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, “Bot-sort: Robust associations multi-pedestrian tracking,” 2022.
- [16] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [17] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, “Omni-scale feature learning for person re-identification,” *arXiv preprint arXiv:1905.00953*, 2019.
- [18] J. Deng, J. Guo, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” *CoRR*, vol. abs/1801.07698, 2018.
- [19] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu, “Cosface: Large margin cosine loss for deep face recognition,” *CoRR*, vol. abs/1801.09414, 2018.
- [20] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, “Dinov2: Learning robust visual features without supervision,” *arXiv preprint arXiv:2304.07193*, 2023.
- [21] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable person re-identification: A benchmark,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1116–1124, 2015.
- [22] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.
- [23] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in *European Conference on Computer Vision Workshops*, pp. 17–35, Springer, 2016.
- [24] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.
- [25] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “VGGFace2: A dataset for recognising faces across pose and age,” in *2018 13th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 67–74, 2018.

- [26] V. Iashin, H. Lee, D. Schofield, and A. Zisserman, “Self-supervised learning on camera trap footage yields a strong universal face embedder,” 2025. ChimpUFE GitHub Repository. Commit <5d79ec9>(Sep 15 2025).
- [27] S. Pertuz, D. Puig, and M. A. Garcia, “Analysis of focus measure operators for shape-from-focus,” *Pattern Recognition*, vol. 46, no. 5, pp. 1415–1432, 2013.
- [28] ismail19797, “Illustration de chimpanzé sur fond blanc.” Illustration vectorielle, ressource no 2433809203 (catégorie: Animaux/Faune sauvage).
- [29] F. Murtagh and P. Contreras, “Algorithms for hierarchical clustering: An overview,” *WIREs Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.
- [30] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
- [31] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.
- [32] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
- [33] Y. Geifman and R. El-Yaniv, “Selective classification for deep neural networks,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [34] T. D. Pereira, N. Tabris, A. Matsliah, D. M. Turner, J. Li, S. Ravindranath, E. S. Papadoyannis, E. Normand, D. S. Deutsch, Z. Y. Wang, G. C. McKenzie-Smith, C. C. Mitelut, M. Diez Castro, J. D’Uva, M. Kislin, D. H. Sanes, S. D. Kocher, S. H. Wang, A. L. Falkner, J. W. Shaevitz, and M. Murthy, “SLEAP: A deep learning system for multi-animal pose tracking,” *Nature Methods*, vol. 19, no. 4, pp. 486–495, 2022.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl